

Treball de Fi de Grau

## **Grau en Enginyeria en Tecnologies Industrials**

Disseny d'un cercador per descobrir i localitzar informació en  
l'entorn corporatiu de l'Institut Municipal d'Informàtica de  
l'Ajuntament de Barcelona

### **MEMÒRIA**

**Autor:** Josep Solé Vives  
**Director:** Lluís Pérez Vidal  
**Convocatòria:** Setembre 2016



Escola Tècnica Superior  
d'Enginyeria Industrial de Barcelona





## Abstract

This project has been motivated by the necessity of launching a tool that allows the more than eleven thousand employees of the Barcelona City Council to locate and especially to find out information that is available and may be helpful for them but so far has been difficult to access. Taking as starting point such problem, it has been designed a search engine that by tracking the user's inquiries, gives back those messages that have happen to be partially or completely matching.

However, the aim of the tool goes beyond. In order to make it more responsive and user-friendly, a number of features have been implemented. The method consist of an easy design of the searcher, the non-return of silences and take advantage of metadata left by the users to improve their experience in future searches. The result is a product that allows to find records either in a direct way, in case the search has been successful, or indirectly, in case there has not been matching results at first, because it gathers and filters the user's data through a number of classifications that improve the search for the next time.

## Resum

Aquest projecte ha estat motivat per la necessitat de dissenyar una eina que permetés als més d'onze mil treballadors de l'Ajuntament de Barcelona localitzar, i el que és més important, descobrir informació de la qual disposen però que, a vegades, no saben ni que existeix. Amb aquesta premissa, s'ha dissenyat un cercador que, mitjançant el text escrit per l'usuari, retorna aquells registres de la informació que han resultat coincidents parcial o totalment.

Però la missió d'aquesta eina no acaba aquí. Per tal d'adaptar-la millor als seus usuaris, s'han implementat una sèrie de funcions amb aquest objectiu. El mètode és que el cercador tingui un disseny fàcil, no retorni silencis i reaprofiti les metadades que deixen els usuaris per a millorar l'experiència en futures cerques. El resultat n'és un producte que permet trobar registres de manera directa, si la cerca ha tingut èxit, i de manera indirecta, quan no ha trobat cap coincidència en primera instància, ja que dona la possibilitat a l'usuari de navegar per una sèrie de classificacions que permeten filtrar la informació per a tenir èxit en la segona cerca.



# Índex

<b>ABSTRACT</b>	<b>1</b>
<b>ÍNDEX</b>	<b>3</b>
<b>1. PREFACI</b>	<b>5</b>
1.1. Origen del projecte .....	5
1.2. Motivació .....	5
<b>2. INTRODUCCIÓ</b>	<b>7</b>
2.1. Objectius.....	7
2.2. Abast .....	7
<b>3. CONEIXEMENTS PREVIS</b>	<b>9</b>
3.1. El codi. Visual Basic .....	9
3.2. Tipus de motors de cerca .....	9
<b>4. ESTRUCTURA DE LA BASE DE DADES</b>	<b>11</b>
4.1. Organització de l'inventari .....	11
4.2. Presentació de les taules de dades .....	14
<b>5. SOLUCIÓ</b>	<b>17</b>
5.1. La interfície .....	17
5.1.1. La interfície en fer la cerca .....	17
5.1.2. La interfície en mostrar els resultats.....	18
5.2. El motor de cerca .....	20
5.2.1. L'ajuda per autocompletar.....	20
5.2.2. La cerca i l'ajuda del Tesauro i la funció de primitivització.....	22
5.3. Presentació dels resultats .....	26
5.3.1. Correspondència dels registres.....	26
5.3.2. La navegació per etiquetes .....	27
5.3.3. La fitxa informativa del registre.....	28
<b>6. ALGORISME EN DETALL</b>	<b>31</b>
6.1. La realimentació .....	31
6.1.1. En què consisteix .....	31
6.1.2. L'actualització de les metadades .....	32
6.1.3. L'ordre d'aparició dels registres .....	33
6.2. L'arbre conductor.....	35
6.3. Els suggeriments .....	40

<b>7. PARAMETRITZACIÓ DE LA INFORMACIÓ</b>	<b>43</b>
7.1. Avantatges de la parametrització.....	43
7.2. Què i com s'ha parametritzat.....	43
<b>8. PLANIFICACIÓ I COST</b>	<b>47</b>
8.1. Planificació.....	47
8.2. Cost.....	48
<b>CONCLUSIONS</b>	<b>51</b>
<b>AGRAÏMENTS</b>	<b>53</b>
<b>BIBLIOGRAFIA</b>	<b>55</b>
Referències bibliogràfiques .....	55
Bibliografia complementària .....	55
<b>ANNEX I: CODI VISUAL BASIC</b>	<b>57</b>
<b>ANNEX II: FITXER “CERCADOR.INI”</b>	<b>98</b>

# 1. Prefaci

## 1.1. Origen del projecte

Aquest projecte ha estat realitzat durant l'estada de pràctiques curriculars a l'Institut Municipal d'Informàtica de l'Ajuntament de Barcelona (IMI). La idea del projecte va ser del tutor a l'entitat, l'Isaac Aparicio. Feia temps que Aparicio tenia una idea per un projecte intern que consistia en satisfer una necessitat que, actualment, en aquesta era de la sobrecàrrega d'informació, esdevenia important. Consisteix en dotar els més d'onze mil treballadors de l'ajuntament d'una eina que els permeti accedir, localitzar, descobrir o, ni que sigui, trobar alguna pista que els ajudi a poder fer ús de manera activa de l'enorme quantitat d'informació de què disposa l'Ajuntament de Barcelona.

Facilitant l'accés a la informació (sovint a informació totalment desconeguda pels empleats de l'ajuntament pel fet que l'ajuntament consta de molts departaments ubicats per bona part de la ciutat i que treballen de manera independent els uns dels altres), es pretén optimitzar els esforços i això, de ben segur, servirà per interconnectar millor aquests diferents departaments.

A més, d'aquesta manera es pot millorar la qualitat de futurs projectes, ja que facilitar l'accés a la informació i descobrir-ne de nova permetrà dotar de millores qualitatives i, tan de bo, evitar actuals maldecaps per no saber a qui o a on anar a buscar certa informació o certes dades necessàries en el dia a dia de bona part d'aquests milers d'usuaris de l'ajuntament.

En resum, calia dissenyar un cercador que entri en una base dades (que prèviament s'ha de configurar), trobi coincidències amb allò que l'usuari hagi escrit i entregui una fitxa amb informació detallada sobre el que busca.

## 1.2. Motivació

Les motivacions per endinsar-se en aquest projecte són múltiples. En primer lloc, una motivació és la oportunitat de realitzar un pla d'aquesta magnitud, que serà utilitzat i ajudarà a persones i no quedarà simplement en un calaix. En segon lloc, per poder tenir una formació professional en el camp de la informàtica que pot ser de gran utilitat en un futur. Finalment, també per les ganes de correspondre de la millor manera possible a aquesta responsabilitat i

per realitzar el primer projecte professional amb èxit.



## 2. Introducció

### 2.1. Objectius

Com s'ha comentat al punt 1.1. *L'origen del projecte*, la finalitat del projecte és l'elaboració d'una eina que satisfaci la necessitat d'introduir la informació de què disposa l'Ajuntament de Barcelona als seus usuaris.

No obstant això, es vol que aquesta eina (d'ara endavant, cercador), a més, sigui proactiva fent la seva funció. Es desitja que l'usuari pugui no només localitzar la informació sobre allò que busca, sinó que sigui capaç de descobrir-ne de nova. En definitiva, es vol trobar una solució al següent diàleg, que descriu perfectament la problemàtica actual:

- Vull saber de quina informació es disposa, on es localitza, etc.
- Molt bé, pregunta.
- ... -no sap què preguntar-

A més, també es vol que l'eina recuperi tota la informació possible sobre què busquen els usuaris, quines cerques condueixen a quins registres i si amb la cerca executada s'ha trobat allò que es desitjava. Per tant, es pretén dissenyar un algorisme que permeti aprofitar aquestes metadades que deixen els usuaris en cada cerca per poder millorar les futures cerques i enriquir-ne contínuament l'experiència.

### 2.2. Abast

L'abast del projecte inclou els següents punts:

- L'estudi de la informació que hi ha a les bases de dades
- L'estructuració i organització de la base de dades per tal de poder mantenir amb èxit la informació que anirà actualitzant-se contínuament. Cal remarcar la importància que té una organització neta i fàcil per tal de fer un manteniment adequat de la base de dades. Al cap i a la fi és la font on el cercador ha d'anar a buscar la informació.

- El disseny d'un motor de cerca que, realimentat amb la informació extreta dels interessos dels usuaris, evolucioni per donar contínuament una millor experiència ús rere ús.
- La confecció d'un arbre conductor que permeti als usuaris que no saben què buscar o als que simplement volen descobrir nova informació sense preguntar, navegar per unes classificacions que aglutinin els registres i que, per tant, serveixin de filtre.
- La parametrització de variables per tal de poder alterar consultes sobre la base de dades o la modificació de camps sense haver d'entrar al codi. D'aquesta manera es pot fer un manteniment de l'eina molt més eficient, ja que fer petites modificacions del codi per persones diferents cada vegada suposa una dificultat afegida que es resol amb la parametrització.
- Una interfície neta, clara i fàcil perquè l'usuari s'adapti el més ràpid possible i no utilitzi l'eina amb la por de no saber-la fer funcionar.

### 3. Coneixements previs

Abans de començar a escriure el cos del codi, s'han hagut de resoldre una sèrie de tasques prèvies. És molt ineficient començar a programar sense tenir clara l'estructura, quines necessitats cal satisfer, com serà la informació que es consulta o quin serà el llenguatge que s'utilitzarà.

Prèviament s'han analitzat els objectius i s'ha definit l'abast del projecte, on s'han concretat les principals necessitats que ha de satisfer el cercador. Encara, però, cal definir l'organització de la base de dades, l'estructura del motor de cerca i el llenguatge de programació que s'usarà.

#### 3.1. El codi. Visual Basic

El llenguatge de codi més utilitzat a l'*Institut Municipal d'Informàtica* per programar softwares interns és el Visual Basic. Tot i tenir coneixements previs de C++, s'ha decidit seguir amb el llenguatge que es fa servir a l'IMI ja que, a part de ser el més coherent (és el llenguatge més comú usat per programar en aquesta empresa), també és el llenguatge amb què els companys s'hi senten més còmodes i, en el cas de necessitar alguna ajuda puntual, facilita poder rebre un cop mà. A més, és un llenguatge molt fàcil d'aprendre si un ja coneix un altre llenguatge de programació, amb la qual cosa, el temps d'aprenentatge de Visual Basic no requereix, ni molt menys, el temps que es necessitaria per aprendre'l si es partís de zero en programació.

Cal afegir que abans de començar aquest projecte, ja s'havia mantingut contacte amb aquest llenguatge en un altre petit projecte que consistia en redissenyar la interfície i afegir-hi alguna funció addicional d'un software propi de l'IMI d'exploració de bases de dades. Per últim, a més d'aprendre a la perfecció el codi esmentat, també ha calgut fer un esforç per aprendre a realitzar consultes SQL a les bases de dades.

#### 3.2. Tipus de motors de cerca

Els cercadors, motors de cerca o, com se'ls anomena tècnicament, Sistemes de Recuperació

d'Informació<sup>[1]</sup>, són eines que, a partir d'una consulta, són capaces d'escollir dins d'una gran col·lecció de documents aquells que corresponen a la consulta i d'ordenar-los segons la seva rellevància.

A dia d'avui, ningú pot posar en dubte la importància que tenen a escala mundial. Cercadors de la magnitud de *Google* o *Bling* processen milions de documents en cada cerca i estan dotats d'algorismes molts complexos que els permeten tenir una gran intel·ligència artificial. La intel·ligència artificial està dividida en nombroses branques, com per exemple el *Reconeixement de Formes*, la *Robòtica*, la *Visió Artificial* o el *Processament del Llenguatge Natural*. I, precisament, els Sistemes de Recuperació d'Informació, són una aplicació d'aquesta última branca de la intel·ligència artificial esmentada, el *Processament del Llenguatge*.

La manera d'operar d'aquests cercadors, a *grosso modo*, és comptar el nombre de vegades que apareix cada paraula cercada i, a partir d'aquest valor, es calcularà la importància de la paraula en el document i es podrà definir quin document té més probabilitats de ser més rellevant que un altre. Alhora, se sol calcular la posició de la paraula en el document, ja que si les paraules que es busquen es troben més juntes serà més probable que aquell document sigui d'interès per a la consulta.

En el cas d'aquest projecte, no s'ha d'entrar a buscar paraules en documents, la qual cosa facilita notablement el disseny del motor de cerca. Com es veurà en el proper punt, el lloc on s'ha de fer la consulta és una taula confeccionada amb informació de la base de dades de què disposa l'IMI i on cada fila correspon a un registre. La informació que cal retornar a l'usuari és, d'una banda, saber què existeix sobre la consulta efectuada i, d'una altra, informació sobre el lloc on es troba, qui n'és el responsable o un telèfon de contacte, entre d'altres.

Com s'explicarà més endavant, concretament en el punt 4, el tipus de cercador que s'ha dissenyat és un cercador per etiquetes. Això vol dir que aquesta eina cerca en un camp de la taula on s'hi troben tota una sèrie de paraules clau (etiquetes) que tenen la missió de descriure el contingut de cada registre.

---

<sup>[1]</sup> Definició extreta del web: <http://www.humanlevel.com/articulos/desarrollo-web/como-functiona-un-buscador-como-google-indexacion.html>



## 4. Estructura de la base de dades

### 4.1. Organització de l'inventari

Aquest és el primer gran assumpte que s'ha hagut de tractar. De la manera d'organitzar la informació depèn que aquest projecte pugui arribar a bon port i que tingui continuïtat o que en la següent actualització s'hagi de refer gran part del producte i, per tant, deixar en evidència l'èxit o el fracàs del projecte. És estrictament necessari que la taula on s'emmagatzema la informació de l'inventari sigui clara, entenedora i fàcil d'ésser actualitzada per a què aquest producte pugui prosperar i tenir continuïtat.

A continuació s'explicarà breument com era la base de dades de què es va partir i s'il·lustrarà la seva evolució fins l'estructura final. Primerament, es disposava d'un document *Excel* que tenia varies fulles de càlcul repletes d'informació com la que es mostra a la *Figura 1*.

	A	B	C	D	E	F	G	H	I	J
1	Grup de dades	Concepte								
2	Elements Inials	Ayuntamiento T								
3	Elements Inials	Barreras de Seguridad								
4	Elements Inials	Barreras de Seguridad - Biondas								
5	Elements Inials	Inox								
6	Elements Inials	Jardin Juegos Infantiles								
7	Elements Inials	Junta Elastoplástico								
8	Elements Inials	Junta Metálico								
9	Elements Inials	Junta Neoprén Armat								
10	Elements Inials	Junta Neoprén Comprimido								
11	Elements Inials	Junta Tall i Segellat								
12	Elements Inials	Otras								
13	Elements Inials	Pasamanos								
14	Elements Inials	Protectores Kioskos, etc.								
15	Elements Inials	Rejilla								
16	Elements Inials	Tubo Cilindrico en Línea								
17	Elements Inials	Tubo Cilindrico Modular								
18	Elements Inials	Tubo Cuadrado								
19	Elements Inials	Túneles y Rondas								
20	Horizontal - Grup marca vial	Área Verda / Blanca								
21	Horizontal - Grup marca vial	Área Verda Exclusiva								
22	Horizontal - Grup marca vial	Bandas Sonoras								
23	Horizontal - Grup marca vial	BCN1.10								
24	Horizontal - Grup marca vial	BCN1.3								
25	Horizontal - Grup marca vial	BCN1.7.1								
26	Horizontal - Grup marca vial	BCN1.7.2								
27	Horizontal - Grup marca vial	BCN2.1								
28	Horizontal - Grup marca vial	BCN2.3								

Figura 1: Excel amb tota la informació de què es disposava quan es va iniciar el projecte.

D'aquest document es va evolucionar cap a un que englobés tota la informació que es va creure rellevant en una sola fulla de càlcul. El resultat és el següent:

	A	B	C	D	E	F
1	ener	f	Concepte	Contingut	Detall	
2	1000	1	Estoo Llicències	Activitats, Espai públic, Expedients d'edificació, Obres, Uso	tècnica, Obres majors, Obres menors, Primera	Llicències - Ecologia Urbana
3	2000	2	Graf Vial	Trams, Nusos	esq, Codis de carrer, Lloc	IMI - IdBC
4	3000	3	Estudis Opinió	Mapes de valoració, Enquesta Serveis Municipals	Activitats culturals, Aparcament, Asfaltat,	IMI - IdBC - BiMap
5	4000	4	Tambor	Projectes de ciutat	electoral, Complexitat, Transversalitat	IMI - IdBC - BiMap
6	5000	5	Punts Calents	Punts conflictius	pública, Drogues, Venda ambulat, Valoració,	IMI - IdBC - BiMap
7	6000	6	Soroll	Demandes Guàrdia Urbana, Denúncies Guàrdia Urbana, Que	Bars de copes, Centres de culte, Decibels, Acústio	IMI - IdBC - BiMap
8	7000	7	Estoo Llicències	Activitats, Espai públic, Expedients d'edificació, Obres, Uso	tècnica, Obres majors, Obres menors, Primera	IMI - IdBC - BiMap
9	7000	8	Llicències	Altes de llicències, Sortides de llicències, Activitats, Obres,	tècniques de projectes, Traspassos, Informes	IMI - IdBC - BiMap
10	7000	9	Terrasses i Velladors	Autoritas, Activitats		IMI - IdBC - BiMap
11	7000	10	Qualitat de Vida, Igualtat i Esports QVIE	Expedients Dependència, Visites domiciliàries, PIA		IMI - IdBC - BiMap
12	7000	11	Habitatges d'ús turístic	Parcel·les amb expedients, Autoritas		IMI - IdBC - BiMap
13		12	Activitats, Pla d'usos			IMI - IdBC - BiMap
14		13	Assentaments i Ocupacions	Assentaments irregulars, Ocupacions		IMI - IdBC - BiMap
15		14	Assoliments	Tambor		IMI - IdBC - BiMap
16		15	Cultura, Educació	Equipaments ASIA, Centres Cívics, Casals de barri, Museus, Casals i Espais Joves, Biblioteques, Sales d'Art E		IMI - IdBC - BiMap
17		16	Punts calents	Prostitució, Soroll, Sobreocupació de la via pública, Drogues, Venda ambulat		IMI - IdBC - BiMap
18	8000	17	CiesServeis_AGBAR		Cal omplir dades de cerca	IMI - IdBC - CiesServeis
19	9000	18	CiesServeis		Cal omplir dades de cerca	IMI - IdBC - CiesServeis
20	10000	19	CiesServeis_GasNatural		Cal omplir dades de cerca	IMI - IdBC - CiesServeis
21	11000	20	CiesServeis_Telefònica		Cal omplir dades de cerca	IMI - IdBC - CiesServeis
22	12000	21	CiesServeis_CLABSA		Cal omplir dades de cerca	IMI - IdBC - CiesServeis
23	13000	22	Xarxa Fibra IMI	Fibra i Wifi	Cables, Fibres i Serveis, Punts Wifi	IMI - Telecomunicacions
24	14000	23	Enllumenat		Suports, Armaris, Cables	Manteniment i Serveis - Ecologia Urbana
25	15000	24	Pavimentació	Polígons de calçada	Tram, Subtram, Plataforma única, Calçada, Vorerà	Manteniment i Serveis - Ecologia Urbana
26	16000	25	Zones Subvenció Paisatge Urbà	Polígons d'afectació		IMPU - Institut Municipal del Paisatge Urbà
27	17000	26	Zones Pla d'Usos de Ciutat Vella	Polígons d'afectació		Districte 01 - Ciutat Vella
28	18000	27	Àrees Districtes Postals	Polígons d'afectació		IMI - IdBC

Figura 2: Excel on s'ajunta tota la informació reestructurada en la fulla de càlcul "UNIO".

Es pot observar que s'han acolorit fulles de càlcul i algunes cel·les, però no és rellevant, ja que es va fer només per facilitar la visualització d'allò que estava fet o que estava per acabar, per exemple.

A partir d'aquest Excel, es va arribar a un primer acord de distribuir la informació en tres columnes, que serien Tema, Subtema i Detall, de manera que cada fila pertanyeria a un registre i cada registre quedaria descrit per aquestes tres columnes. Per tant, calia encara fer un esforç considerable i organitzar tota aquesta informació per tal que es tinguessin les tres columnes farcides de contingut i es pretenia que el cercador cerqués a les tres columnes.

Tanmateix, es va observar que era una feina feixuga i que dificultava molt la claredat dels registres, ja que no sempre es podien omplir les tres columnes. A part, també complicava l'estructura i, conseqüentment, no complia amb l'objectiu de crear una taula de dades de fàcil manteniment. A més, en una primera versió pilot del cercador, sorgien moltes dificultats degudes a tenir els registres estructurats d'aquesta manera i d'haver de recórrer les tres columnes. En molts casos i com a mostra que aquesta proposta no anava ben encaminada, diferents registres contenien informació molt semblant però en columnes diferents i provocava un cert caos que calia resoldre.

La solució, com s'ha vist quan s'han estudiat els tipus de motors de cerca existents, ha estat la d'organitzar la informació per tal de poder cercar per etiquetes. Això consisteix en fer que cada registre, d'una banda, tingui una part descriptora de la informació continguda (el registre en sí) i, de l'altra, un camp que es farceix amb totes aquelles etiquetes que descriu el registre i la informació que conté. En aquest camp és on únicament el cercador ha de buscar les coincidències.

D'aquesta manera, finalment s'ha decidit que la informació associada a la descripció del registre es dividirà en dos camps enlloc dels tres anteriors i seran Tema, que com es veurà més endavant serà la qüestió genèrica a què fa referència el registre, i Subtema, que serà la descripció concreta del registre. El tercer camp es dedicarà exclusivament a l'esmentat farciment d'etiquetes.

D'aquesta manera, registres que acaben referenciant a la mateixa font d'informació, tal i com s'observa en la *Figura 3*, s'han pogut ajuntar. Així, el camp *Concepte*, on cada registre tenia una descripció diferent però un mateix *pare* ("Elements lineals"), passa a ser el camp *Etiquetes*; alhora que tots els registres "Elements lineals" passen a ser-ne un de sol. En resum, en aquest cas d'exemple, l'antic *Concepte* passa a agrupar-se i formar el conjunt d'etiquetes que descriuen el contingut d'aquest registre, a les quals s'hi afegixen, també, etiquetes descriptores del registre (*Tema* i *Subtema* com a etiquetes).

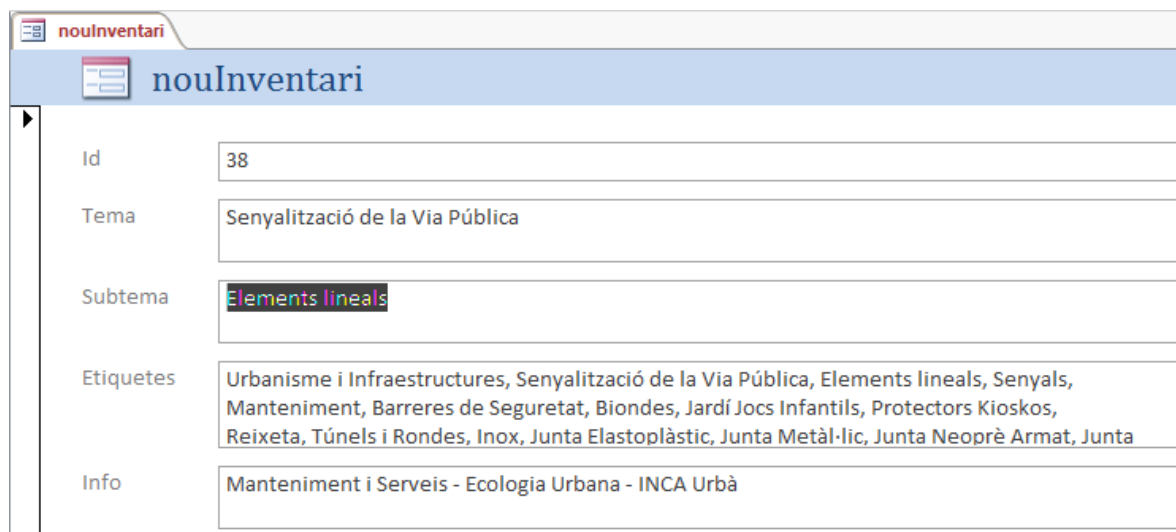
És a dir, s'ha fet una conversió *n a 1* dels registres que referenciaven a un mateix concepte, que en aquest exemple ha estat "Elements lineals".

La nova taula de dades, després de tota aquesta evolució, ha quedat de la següent manera:

nouInventari				
Id	Tema	Subtema	Etiquetes	Info
31	Urbanisme i Infraestructures	Àrees Bàsiques de Salut - Polígons d'afectació	Urbanisme i Infraestructures, Àrees Districtes Postals, Polígons d'afectació	IMI - IdBC
32	Urbanisme i Infraestructures	Àrea Verda - Polígons d'afectació	Urbanisme i Infraestructures, Àrea Verda, Aparcament, Residents, Polígons d'afectació	Oficina Àrea Verda - BSV
33	Senyalització de la Via	Tipus de senyalització	Urbanisme i Infraestructures, Senyalització de la Via Pública, Senyals, Tipus de senyalització	Manteniment i Serveis -
34	Senyalització de la Via	Tipus de suport	Urbanisme i Infraestructures, Senyalització de la Via Pública, Senyals, Tipus de suport	Manteniment i Serveis -
35	Urbanisme i Infraestructures	Elements del Mobiliari Urbà	Urbanisme i Infraestructures, Elements del Mobiliari urbà, Tipus d'element, Elements del Mobiliari Urbà	Manteniment i Serveis -
36	Urbanisme i Infraestructures	Reserva de Mobiliari Urbà	Urbanisme i Infraestructures, Reserva de Mobiliari urbà, Tipus de reserva, Bili	Manteniment i Serveis -
37	Senyalització de la Via	Grup de senyals	Urbanisme i Infraestructures, Senyalització de la Via Pública, Senyals, Grups de senyals	Manteniment i Serveis -
38	Senyalització de la Via	Elements lineals	Urbanisme i Infraestructures, Senyalització de la Via Pública, Elements lineals	Manteniment i Serveis -
39	Senyalització de la Via	Elements Horitzontals	Urbanisme i Infraestructures, Senyalització de la Via Pública, Elements Horitzontals	Manteniment i Serveis -
40	Urbanisme i Infraestructures	Alçada edificis	Urbanisme i Infraestructures, Alçada edificis	IMI - IdBC
41	Seguretat	Accidents gestionats per la Guàrdia Urbana	Seguretat, Accidents, Xocs, Guàrdia Urbana	Guàrdia Urbana
42	Seguretat	Denúncies Guàrdia Urbana	Seguretat, Multes, Denúncies Guàrdia Urbana	Guàrdia Urbana
43	Seguretat	Demandes Guàrdia Urbana	Seguretat, Demandes Guàrdia Urbana	Guàrdia Urbana
44	Gestió d'Incidències	Queixes IRIS	Gestió d'Incidències, Queixes IRIS, Reclamació, Al·legació, Instància, Incidències	IRIS
45	Inspeccions	Autoritas	Inspeccions, Autoritas, Expedient Electrònic d'Inspecció, Tramitació Electrònica	Inspecció - Ecologia Urbana
46	Senyalització de la Via	Plaques Vials	Urbanisme i Infraestructures, Senyalització de la Via Pública, Plaques Vials, R	IMI - IdBC

*Figura 3: Taula de dades final exportada a Access.*

Es pot observar com tots els registres “Elements lineals” s’han agrupat en un de sol. A més, se li ha afegit un *Tema* més genèric (un pare) per tal de classificar-lo millor. Tots els conceptes detallats que tenia cada registre en les versions anteriors han passat a omplir el camp *Etiquetes*. Com que no s’arriba a apreciar a la imatge anterior, es mostrarà un formulari d’Access a continuació on es veurà com després d’“Elements lineals” ve tota la llista de conceptes que abans formaven un registre de sol cadascun d’ells.



The screenshot shows the 'nouInventari' Access form. The form has a blue header with the 'nouInventari' logo and title. Below the header, there is a vertical list of fields on the left and their corresponding input boxes on the right. The fields are: Id (38), Tema (Senyalització de la Via Pública), Subtema (Elements lineals), Etiquetes (Urbanisme i Infraestructures, Senyalització de la Via Pública, Elements lineals, Senyals, Manteniment, Barreres de Seguretat, Biondes, Jardí Jocs Infantils, Protectors Kioskos, Reixeta, Túnel i Rondes, Inox, Junta Elastoplàstic, Junta Metàl·lic, Junta Neoprè Armat, Junta), and Info (Manteniment i Serveis - Ecologia Urbana - INCA Urbà).

Id	38
Tema	Senyalització de la Via Pública
Subtema	Elements lineals
Etiquetes	Urbanisme i Infraestructures, Senyalització de la Via Pública, Elements lineals, Senyals, Manteniment, Barreres de Seguretat, Biondes, Jardí Jocs Infantils, Protectors Kioskos, Reixeta, Túnel i Rondes, Inox, Junta Elastoplàstic, Junta Metàl·lic, Junta Neoprè Armat, Junta
Info	Manteniment i Serveis - Ecologia Urbana - INCA Urbà

Figura 4: Visualització del registre “Elements lineals” en el formulari d’Access on es pot veure com en el camp *Etiquetes* hi ha tota la llista de conceptes que compartien “pare” en les versions anteriors.

Finalment, a la *Figura 3* també s’observa un darrer camp, *Info*, que descriu la fila de la taula “INFORMACIO” a la qual s’han d’anar a buscar les dades informatives que es mostraran en pantalla a l’usuari.

## 4.2. Presentació de les taules de dades

Per a l’elaboració del Cercador es necessiten, a part de la informació descrita en el punt anterior, una sèrie de taules que han d’importar les seves dades a l’algorisme. Totes elles provenen d’una base de dades *Oracle* o, com en la *Figura 5*, d’*Access*.





Figura 5: Imatge extreta de la base de dades d'Access que conté les 6 taules de dades amb les quals opera el Cercador.

A continuació, es descriuen les 6 taules mostrades en la figura anterior:

- “NOUINVENTARI”: En el punt anterior s’ha descrit com s’organitza l’inventari dins la taula de dades d’Access “NOUINVENTARI”. En el codi, aquesta taula s’ha anomenat *taulaInventari*.
- “LLISTATETIQUETES”: Engloba tota una llista de les etiquetes més rellevants. S’utilitzarà per farcir la funció que s’explicarà més endavant per ajudar l’usuari a completar la seva cerca mentre escriu. En el codi es dirà *taulaAutocom*.
- “INFORMACIO”: És la taula on s’emmagatzema tota la informació que se li presentarà a l’usuari cada cop que consulti un registre. En el codi es dirà *taulaInformacio*.
- “OPENDATA”: És una taula que conté els temes genèrics proposats en el servei de dades obertes de l’Ajuntament de Barcelona i que serviran per classificar els registres en grans temes. En el codi es dirà *taulaOpenData*.
- “REALIMENTACIO”: Aquesta taula conté les metadades de les cerques que es fan. Per tant, és una taula que s’actualitza després de cada cerca. En el codi es dirà *taulaRealim*.
- “TESAURUS”: Es tracta d’una taula que relaciona un mot genèric, que en aquest cas és un mot etiqueta, amb paraules equivalents que no són etiquetes pels casos en què un usuari les escrivís. En el codi es dirà *taulaEquivalencies*.



## 5. Solució

En aquest punt es veurà la forma que s'ha donat a la interfície amb la qual hauran d'interactuar els usuaris de l'ajuntament i s'explicarà detalladament el funcionament del motor de cerca.

### 5.1. La interfície

#### 5.1.1. La interfície en fer la cerca

El primer que a hom li ve al cap quan pensa en la interfície d'un navegador segur que és alguna cosa semblant a la pàgina principal de *Google*. I és per aquest motiu que si es vol que un usuari de l'ajuntament que no és expert en ús de softwares pugui fer funcionar fàcilment aquesta eina, la interfície del cercador d'aquest projecte ha de tenir una estructura semblant a l'esmentada.

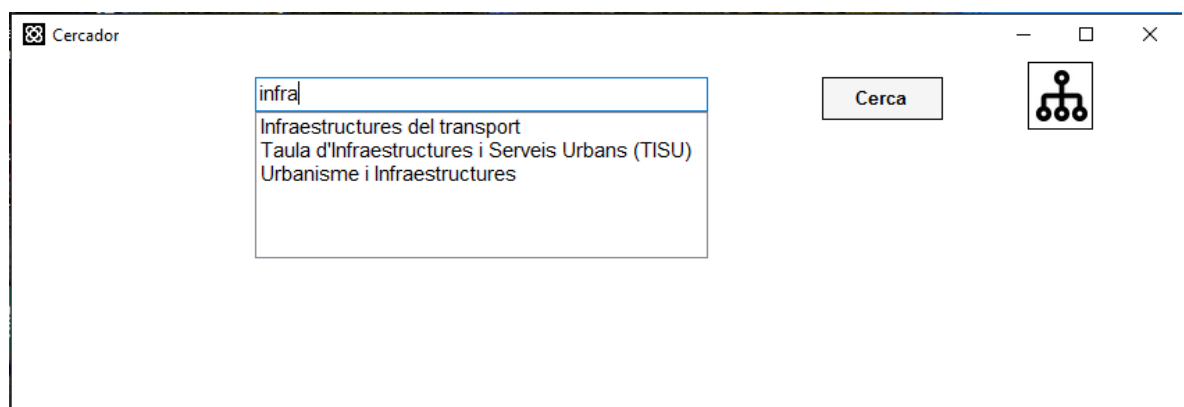


Figura 6: Imatge de la meitat superior de la interfície del cercador.

Com es pot observar consta només de tres elements:

- Un *ComboBox* (*ComboBox1* en el codi), on a la part superior l'usuari escriu allò que vol cercar, i a la part inferior van apareixent suggeriments filtrats cada vegada que s'escriu alguna cosa.
- Un botó de Cerca (*botoCerca* en el codi) que inicia la recerca de coincidències. Teclejar la tecla *Enter* és un equivalent de clicar el botó.

- Un botó que duu al servei d'ajuda (*botoArbre* en el codi) que consisteix en un arbre conductor que ofereix filtratge d'etiquetes segons diverses classificacions de la informació. En passar el cursor per sobre s'ofereix una descripció del que és.

D'aquesta manera, queda clar com s'ha de fer funcionar l'aplicació des del primer moment.

### **5.1.2. La interfície en mostrar els resultats**

El raonament per dissenyar la interfície quan s'han de mostrar els resultats de les coincidències trobades en la cerca segueix el mateix patró que abans. Davant de la pregunta “Com espera un usuari gens expert trobar els resultats en pantalla?” es troba una resposta clara: d'una manera molt semblant a com els mostren els cercadors Google o Bling.

No obstant això, hi ha d'haver per força alguna petita variació, ja que cal mostrar, per a cada registre, el seus Tema i Subtema que el defineixen i les etiquetes associades. Així, s'ha decidit que la interfície mostri la llista de resultats de la següent manera:

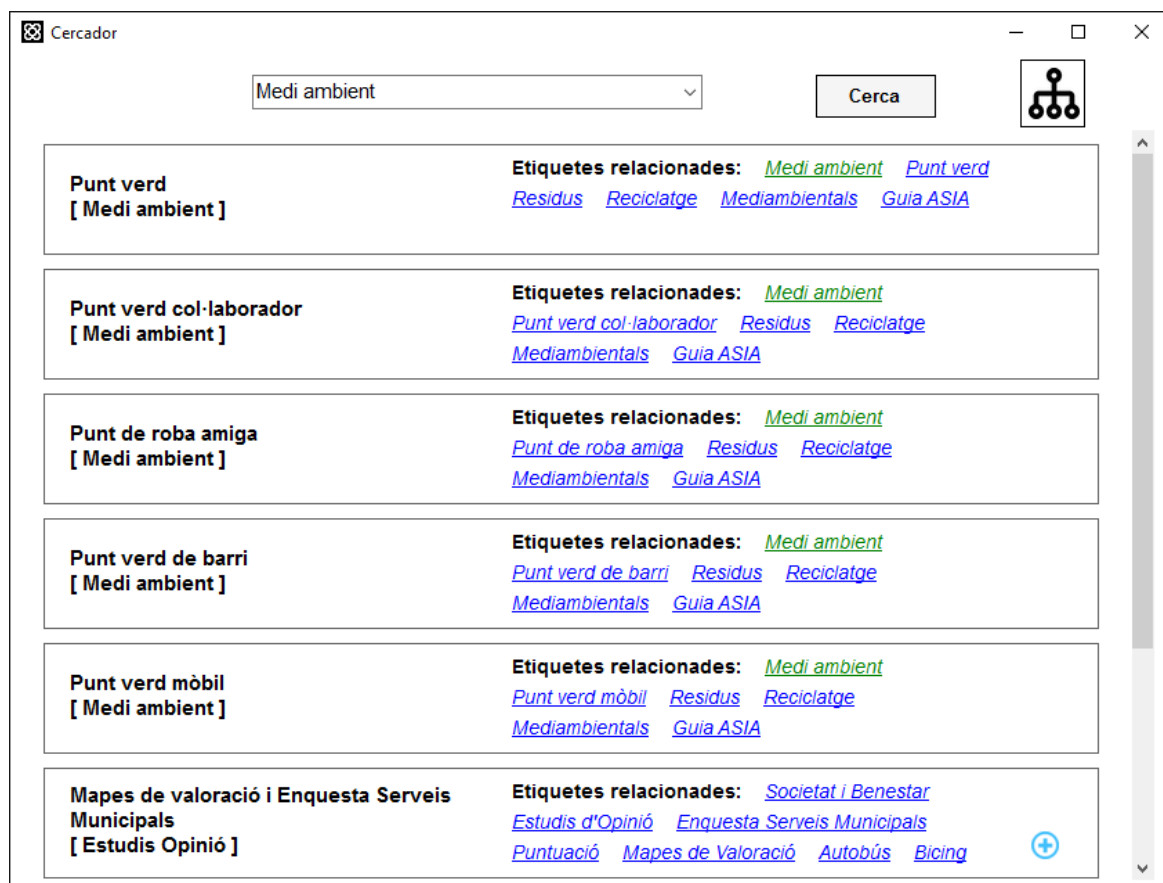


Figura 7: Interfície mostrant els resultats de la cerca "Medi ambient".

Tal i com es pot observar en la imatge anterior, els registres es presenten un sobre l'altre, començant l'ordre d'importància des de dalt. D'una banda, a la part esquerra de cada registre, apareix un botó que s'activa en passar el cursor per sobre d'on apareixen els *Tema* i *Subtema* que descriuen el registre. D'altra banda, a la dreta, apareix una caixa que conté les etiquetes que descriuen el registre i llur contingut.

A més, s'observa com l'etiqueta mitjançant la qual el registre ha estat seleccionat s'acoloreix de color verd com a símbol de coincidència. També s'adverteix com, en el registre que queda a la part inferior de la imatge, apareix una icona "+" de color blau que indica que encara queden etiquetes per mostrar. Només cal fer clic a la imatge per expandir la caixa. Tot això es pot veure en la següent imatge:

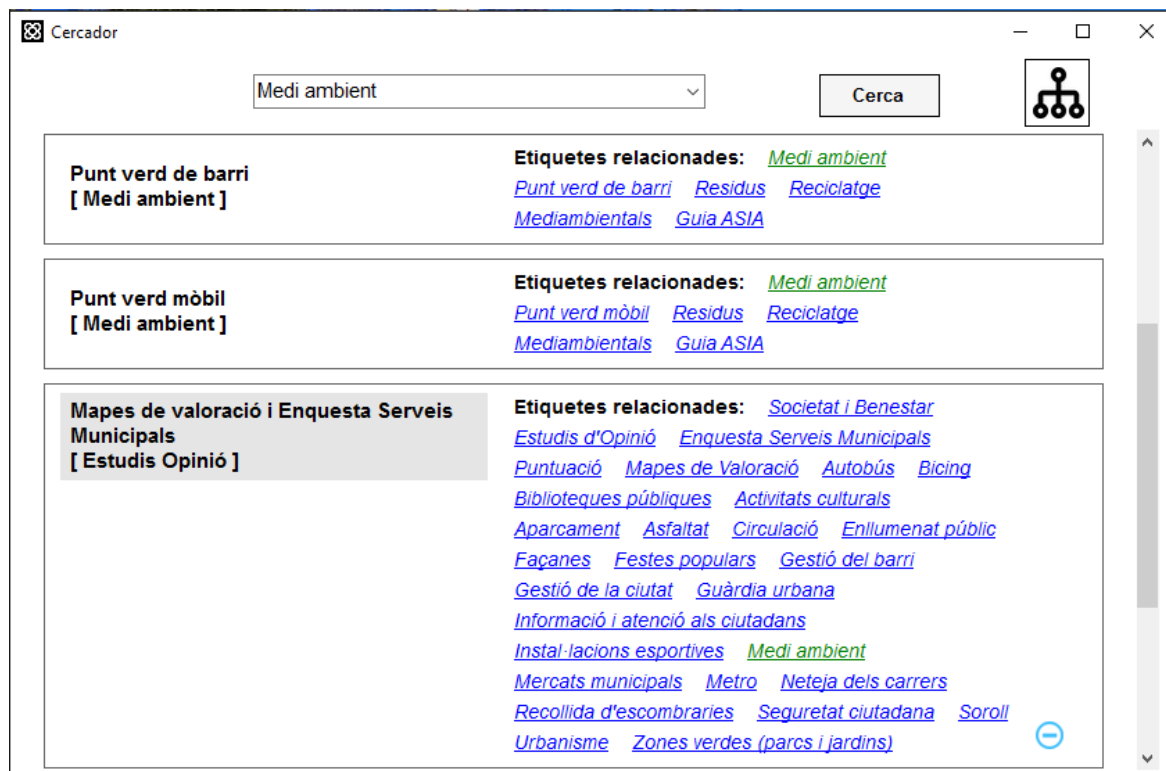


Figura 8: La interfície en el moment en què s'ha expandit una caixa d'etiquetes i el cursor està sobre el botó "Mapes de valoració i Enquesta Serveis Municipals".

S'observa com ara apareix una icona "-" a la part inferior de la caixa expandida i que el botó del registre sobre el qual s'ha situat el cursor queda ombrejat per informar visualment a l'usuari que fent clic obtindrà la fitxa informativa d'aquest registre.

## 5.2. El motor de cerca

Aquí s'explicarà amb detall tot el funcionament del motor de cerca, des del moment en què l'usuari comença a teclejar fins que es mostren els resultats en pantalla.

### 5.2.1. L'ajuda per autocompletar

La idea és que cada cop que l'usuari tecleja una lletra, a la caixa inferior de la barra on s'està escrivint apareguin suggeriments que continguin parcial o totalment allò que està escrit a cada instant. Això s'aconsegueix mitjançant un filtre sobre la taula on s'emmagatzema el llistat amb totes les etiquetes disponibles. El filtre és una consulta SQL a la taula de la següent forma:

```
Dim etiq_like() As DataRow
etiq_like = taulaAutocom.Select("Etiquetes LIKE '%" & text & "%'")

For Each dr As DataRow In etiq_like
    ComboBox1.Items.Add(dr("Etiquetes").ToString)

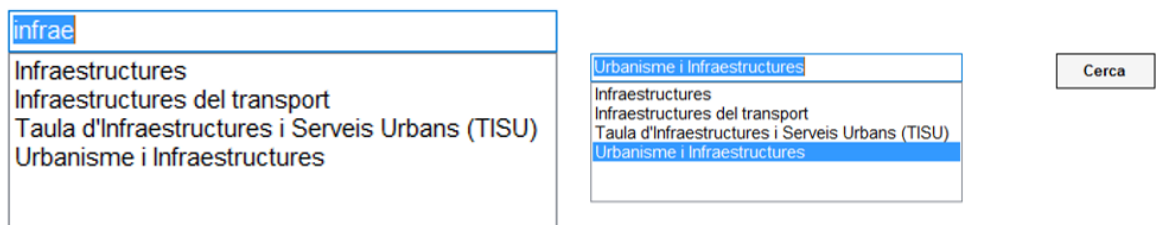
Next
```

*Fragment de codi 1: Es mostra el fragment a partir de qual s'efectua el filtre d'etiquetes per aparèixer a la caixa de suggeriments per autocompletar la cerca.*

En primer lloc cal observar que es crea la variable “etiq\_like()”, que és una fila de dades, és a dir, un vector en el qual en cada posició s’hi guarda un valor.

Després de crear la variable, se li assignen la llista de valors provinents de la taula de dades “taulaAutocom” filtrats per la consulta SQL “Etiquetes LIKE '%" & text & "%'”, on text és una variable creada anteriorment que conté allò que s’està escrivint lliure d’apòstrofs, ja que donen molts problemes degut a que el caràcter apòstrof forma part del llenguatge SQL<sup>[2]</sup>.

Finalment, s’afegeixen tots els elements del vector de dades “etiq\_like()” a la caixa del “ComboBox1”. L’usuari podrà accedir-hi mitjançant les fletxetes del propi teclat o fent-hi clic amb el ratolí.



*Figura 9: A l'esquerra, la llista de suggeriments en el moment en què s'ha escrit “infrae”; a la dreta, com queda seleccionada l'etiqueta “Urbanisme i Infraestructures” després d'haver baixat 4 posicions amb la fletxeta o d'haver-hi clicat a sobre amb el ratolí.*

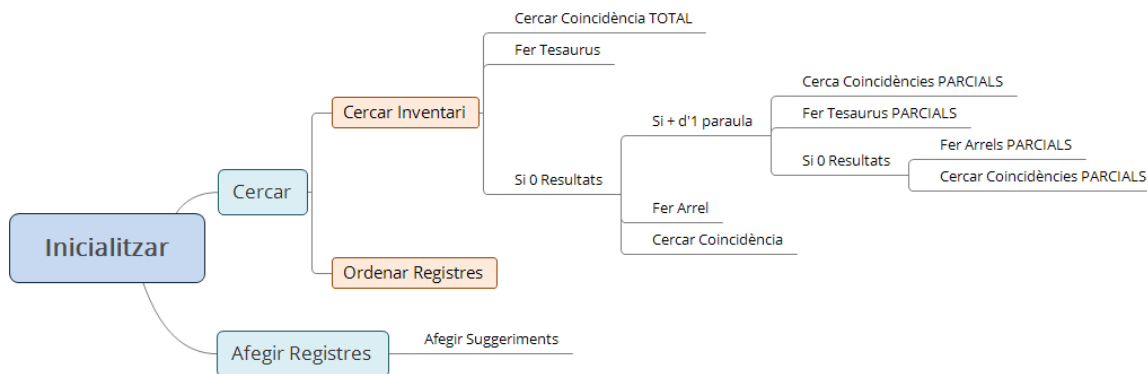
Aquest procediment es repeteix indefinidament cada cop que el text pateix un canvi, és a dir, cada cop que l'usuari escriu o esborra, fins que s'executa la cerca. En l'exemple anterior, *Figura 9*, després de fer la selecció, cal encara executar la cerca o, si es desitja, esborrar i

<sup>[2]</sup> La manera per evitar l'error de lectura en la consulta SQL, s'ha extret del web: <http://stackoverflow.com/questions/1912095/how-to-insert-a-value-that-contains-an-apostrophe-single-quote>

tornar a començar.

### 5.2.2. La cerca i l'ajuda del Tesauros i la funció de primitivització

A partir del moment en què s'executa la cerca comença la part on el motor de cerca ha de trobar coincidències, ordenar-les i mostrar-les en pantalla. A la *Figura 10* es mostra l'esquema del funcionament del motor de cerca.



*Figura 10: Esquema del funcionament del motor de cerca. Amb "TOTAL" es vol dir que se cerquen coincidències amb el que ha escrit l'usuari en bloc, siguin una paraula o vàries. En canvi, amb "PARCIAL" es vol dir que el text escrit es divideix en paraules i el tractament s'efectua individualitzat.*

Cada cel·la de l'esquema que consta d'un verb, pertany a un petit algorisme diferent. A continuació s'explicarà amb deteniment quina és la funció de cada un d'ells:

- **Inicialitzar:** és la funció que es crida cada vegada que s'executa una nova cerca. Abans de passar a *Cercar*, s'encarrega de fer uns petits canvis al text escrit per l'usuari; en primer lloc el normalitza, que vol dir que li treu els accents i dièresis i, en segon lloc, li treu l'apòstrof, ja que, tal i com s'ha explicat amb la funció d'autocompletar, l'apòstrof dóna molts problemes per ser un caràcter propi del llenguatge SQL, amb el qual s'efectuen les consultes.
- **Cercar:** prepara l'escenari per iniciar la cerca. Dóna forma a les taules necessàries si és la primera cerca o les reinicia en cas contrari. Després crida la funció que cerca a l'inventari, "CercarInventari", la qual retorna els resultats definitius i els ordena cridant la funció d'ordenar registres, "OrdenarRegistres", que omplirà de manera



ordenada la taula definitiva.

- **Cercar Inventari:** aquí comença la cerca. El primer pas és el de normalitzar el camp etiquetes de la taula de consulta, de la mateixa manera que s'ha fet en un primer moment amb el text escrit.

A continuació es realitza la comparació entre el text i el camp etiquetes de cada fila de la taula de l'inventari mitjançant la següent comanda:

```
If LCase(" " & etiquetaNorm).Contains(LCase(" " & cercaNorm)) Then
```

*Fragment de codi 2: Condició que indica si una subcadena està continguda dins d'una altra.*

on "LCase" ho passa tot a minúscules per evitar problemes; l'espai evita que una paraula curta pugui aparèixer entremig de l'altra subcadena en la comparativa<sup>[3]</sup> i "Contains" serveix per indicar si una subcadena (text de la cerca) està continguda dins una cadena (camp etiquetes). En el cas que la indicació sigui certa, s'afegeix la fila de dades a la taula *taulaCoincidencies*.

Després de la primera cerca de coincidències, es realitza el Tesaurus.

- **Funció "Tesauro":** consisteix en executar una cerca en una taula que consta de dos camps o columnes: "ETIQUETA" i "EQUIVALENTS"

TESAURUS		
ID	ETIQUETA	EQUIVALENTS
1	Llicències	Licencias
2	Transport	Transports, transportes
3	Aparcament	Aparcaments, Aparcamen
4	Armari	Armaris
5	Esport	Esports, Deporte

*Figura 11: Taula TESAURUS on entra la funció Tesauro a buscar equivalències.*

Mitjançant la comanda "Contains" descrita anteriorment, es busquen coincidències dins del camp "EQUIVALENTS" i, en cas de trobar-ne, s'agafa

<sup>[3]</sup> Aquesta idea s'ha extret de la realització de proves per observar el funcionament de cercadors existents. Concretament, en aquest cas es tracta d'una mancança que es va detectar en el web de l'editorial Lumen en fer la cerca del llibre "Una villa vil" en el seu cercador. El web és el següent: <http://www.edlumen.net>

la paraula corresponent del camp “ETIQUETA”, es torna a executar la cerca i s’afegeix a la taula *taulaCoincidencies* si la nova fila trobada no pertanyia ja a la taula.

En un principi aquesta funció s’havia plantejat com una ajuda extra en cas de no trobar resultats, però després d’una sèrie de proves es va observar que els resultats que es poden proposar amb el Tesauro poden ser d’absolut interès per a l’usuari independentment dels altres resultats obtinguts, de manera que aleshores no s’arribaria a executar mai el Tesauro. Per aquest motiu s’ha decidit que s’executi el Tesauro en cada cerca, independentment de si s’han trobat o no resultats.

Un cop realitzades les dues operacions anteriors, es fa la pregunta: Es tenen zero resultats? En cas afirmatiu, es prossegueix dividint el text en paraules i disposant-les en un vector, una a cada posició. A més, s’eliminen articles i preposicions per tal d’evitar troballes incoherents. Aleshores es repeteixen els dos passos anteriors, cerca i Tesauro, però ara paraula per paraula.

Finalment, si tampoc s’han trobat resultats cercant les paraules per separat o si la cerca ja només constava d’una sola paraula, s’efectua una crida a la funció que primitivitzava cada paraula, la funció “FerArrel”, que retorna la paraula primitivitzada.

- **Funció “Fer Arrel”:** la seva missió és retornar la primitiva de la paraula entrant. Quan es va decidir que es volia fer un algorisme que fes l’arrel d’una paraula, es va realitzar una recerca bibliogràfica per aconseguir un algorisme eficient i bo. Finalment, es va trobar una idea<sup>[4]</sup> a partir de la qual realitzar aquesta eina.

A *grosso modo*, la idea consisteix en extreure la subcadena de caràcters fins la primera, o si es pot, la segona parella (parella inclosa) de vocal + consonant començant des del principi d’una paraula formada per almenys 5 caràcters. Amb el tros de final de mot que s’obté, s’efectua una resta sobre la paraula original, obtenint-ne com a resultat una paraula que comença des del principi, de la mateixa manera que la original, i que acaba en sec just abans d’una



<sup>[4]</sup> Idea extreta d’aquest web: <http://snowball.tartarus.org/algorithms/spanish/stemmer.html>



parella vocal + consonant.

Un cop feta l'arrel de la paraula o de les paraules del vector, es realitza per darrer cop la cerca de coincidències. Si en alguna de les cerques anteriors s'han trobat coincidències, voldrà dir que existeixen resultats i, abans de portar-los a ordenar, cal eliminar-ne els repetits. Fins a aquest punt s'havia evitat que s'afegissin dos registres iguals en la cerca de coincidències per la mateixa paraula. Tanmateix, no s'havien eliminat dos registres iguals si s'havien afegit per tenir coincidències amb dues paraules diferents. I d'aquest fet se n'extraurà profit, ja que com més paraules diferents hagin coincidit en un mateix registre voldrà dir que aquest registre té més possibilitats de ser d'interès per l'usuari que d'altres que tenen un menor nombre de coincidències.

Per tant, en aquesta funció (que consisteix en eliminar repetits), s'aprofita per omplir una columna amb un enter representant el nombre de vegades que s'ha trobat cada registre, al final de la cerca, en la taula de coincidències.

- **Ordenar Registres:** un cop ja es té la llista definitiva de registres, cal establir-ne un ordre i, en alguns casos, filtrar-los.

L'estratègia és, primer, crear una nova taula, *taulaDesordenada*, on s'hi comencen afegint els registres que han obtingut més paraules coincidents (en el cas que s'hagués escrit més d'una paraula) i, a partir d'un cert nombre de registres que tenen més coincidències que d'altres, es pot creure convenient no afegir els que en tenen menys. El nombre s'ha establert que sigui cinc, però es pot modificar sempre que es vulgui quan es cregui convenient.

Posant per cas, si hi ha cinc o més registres amb dues coincidències i vint amb només una, aquests vint ja no s'arribaran a afegir i no es mostraran en pantalla, ja que es creu que aquests cinc o més registres gairebé segur mostraran tot el ventall d'opcions d'interès per l'usuari. En canvi, si hi hagués un registre amb dues coincidències i vint amb una, s'ha establert que es mostrin aquests vint després del registre amb major nombre de coincidències per a què l'usuari pugui tenir un major ventall de possibilitats en el cas que aquest sol registre constés de més coincidències per simple casualitat.

Després d'efectuar aquest filtre per coincidències, s'ordenen els registres segons un coeficient calculat que estableix la prioritat de cada registre segons el text escrit

per l'usuari. Aquest càlcul s'explicarà detalladament en el punt 6.1. *La realimentació.*

### 5.3. Presentació dels resultats

Com s'ha vist i explicat en el punt 5.1.2. *La interfície en mostrar els resultats*, cada registre es mostra com una filera, que es van col·locant un sota l'altre ordenats per prioritat. Per executar la presentació dels resultats, es crida la funció "AfegirRegistres" un cop la funció "Cercar" ha acabat.

#### 5.3.1. Correspondència dels registres

El problema més gran d'aquesta part ha estat com referenciar els objectes que componen cada registre. És a dir, en el moment en què es clica sobre el botó d'un registre, de quina manera es relaciona aquest clic amb la seva referència informativa? O també, en el cas d'expandir la caixa d'etiquetes d'un registre, com saber a partir de quin objecte cal moure cap avall els demés registres?

La solució consisteix en crear unes llistes d'objectes com a variables globals. Cada llista és d'un sol tipus d'objecte. Les llistes en qüestió son:

```
Public llistaPanells As New List(Of Panel)
Public llistaCaixesEtiquetes As New List(Of FlowLayoutPanel)
```

*Fragment de codi 3: Exemples del tipus de variable llista d'objectes. En aquest cas es mostren la llista de panells on cada panell és l'embolcall que conté un registre i la llista de panells d'autoadaptació d'objectes, els quals seran la caixa contenidora de les etiquetes.*

El quid de la qüestió, que és com referenciar cada panell per tal de relacionar-lo amb la informació de la taula, consisteix en anomenar els objectes. Com que els registres estan disposats i ordenats en una taula definitiva, es pot referenciar cada panell amb el número que correspon a la seva posició a la taula i s'afegeix també l'objecte a la llista creada. La manera de fer-ho s'exemplifica així:

```
While i < taula.Rows.Count
    Dim panell As New Panel
    Panel1.Controls.Add(panell)
    With panell
```

```
.Name = i.ToString & "-Panell"
```

*Fragment de codi 4: Mostra del nom que es posa a un objecte "Panel" que es guardarà en una llista de "Panel" per poder ser cridat en un futur.*

Com es pot observar, s'inicia un comptador, es crea el panell que ha de contenir el registre, s'afegeix a la interfície i, tot seguit, s'hi afegeixen una sèrie de característiques de les quals es destaca la que correspon a l'anomenament del panell. D'aquesta manera es podrà anar directament al panell desitjat quan es desitgi. Un exemple és el següent, on es vol cridar el panell sobre el qual l'usuari ha demanat una expansió de la caixa de les etiquetes i, com a conseqüència, cal expandir l'alçada del panell:

```
i = CInt(Microsoft.VisualBasic.Left(sender.Name, nDigits))  
With llistaCaixesEtiquetes.ElementAt(i)  
    .Height = .Controls.Item(.Controls.Count - 1).Location.Y + 20  
End With
```

*Fragment de codi 5: Mostra de la forma amb què es crida l'objecte que fa de caixa d'etiquetes de la llista que el conté i s'hi efectua un canvi en l'amplada.*

S'observa que per cridar l'element "i" de la llista en qüestió cal extreure'n primer el valor de l'enter "i". Per fer-ho s'executa una comanda de Visual Basic que agafa el nombre de caràcters des de l'esquerra que se li digui, que en el cas d'estudi són el nombre de caràcters abans del guió i que ve donat per la variable "nDigits". Posteriorment, s'extreu l'element "i" de la llista i s'hi efectuen les modificacions pertinents.

### 5.3.2. La navegació per etiquetes

La mostra de resultats ofereix una interessant possibilitat, que és la de navegar per etiquetes. Les etiquetes contingudes dins la caixa tenen l'aspecte de *link* perquè ho són. En clicar una etiqueta es realitza una nova cerca amb el text *linkat* seleccionat. Tanmateix, si es decideix retornar a la cerca anterior, només cal clicar la icona que apareix a la part superior de la pantalla quan s'ha fet una nova cerca.

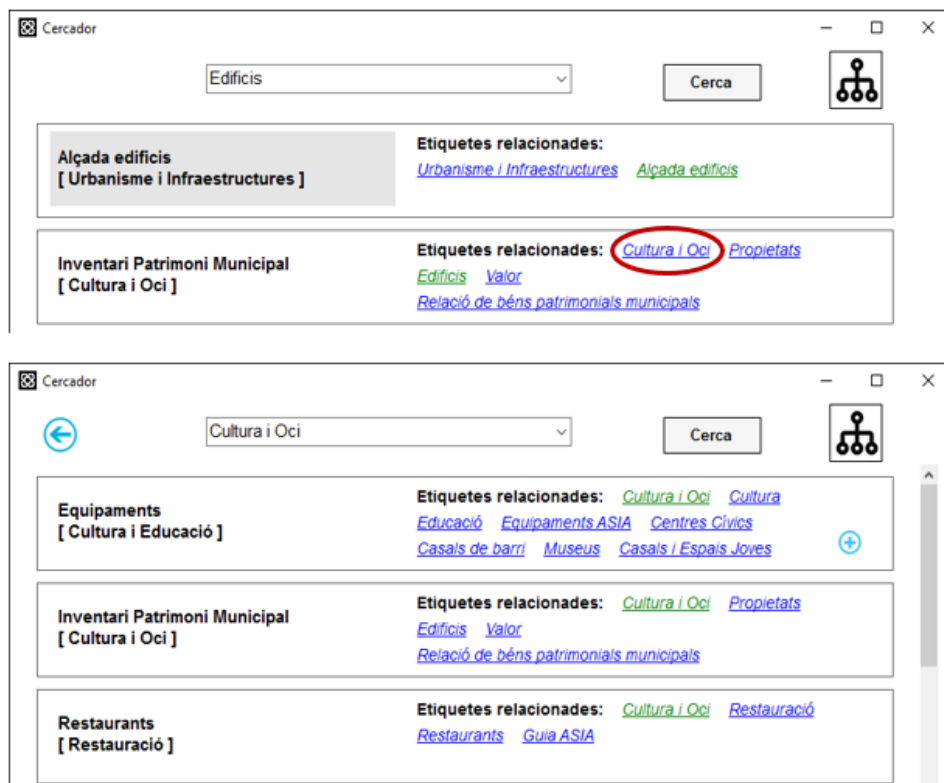


Figura 12: A la imatge superior, s'observa que s'ha cercat "Edificis". A la inferior, la cerca és "Cultura i Oci". Però la segona cerca s'ha executat clicant sobre l'etiqueta "Cultura i Oci" encerclada.

S'observa que a la segona imatge apareix la icona per tornar a la cerca anterior, que seria la pantalla de la imatge superior.

### 5.3.3. La fitxa informativa del registre

Quan l'usuari clica sobre el botó del registre que vol consultar, s'obre una nova finestra que és la fitxa informativa del registre en qüestió. Les fitxes extreuen les dades de la taula "INFORMACIO" de la base de dades i mostren informació de les cel·les no buides de la taula. La taula és la següent:

INFORMACIÓ								
ID	RESULTAT	NOM	ADREÇA	CORREU/WE	TELÈFON	CONTACT	INFORMACIÓ ADDICIONAL	
1	Guàrdia Urbana	Guàrdia Urbana	diverses	www.bcn.cat/	092			
2	Manteniment i Serveis - Ecologia Urbana	BCNecologia, l'Agència de Ecologia Urb	C. Escar 1, 3 · 08039 B	info@bcnecol	93 224 0860			
3	IMPU - Institut Municipal del Paisatge Urbà	Institut Municipal del Paisatge Urbà i C	Av. Drassanes 8		932562525			
4	IRIS	Servei d'Incidències, Reclamacions i S	C/Campins, 24	iris@santcelor	93 864 12 10			
5	Districte 01 - Ciutat Vella							
6	IMI - IdBC - BiMap	Institut Municipal d'Informàtica - Insti	Avinguda Diagonal 22		932918100			
7	Patrimoni Municipal - Gerència de Recursos	Oficina d'Informació d'Oposicions i Cor	C. Escar, 1, La Barcelor	www.bcn.cat/	934027740			
8	IMI - Telecomunicacions	Institut Municipal d'Informàtica - Insti	Avinguda Diagonal 22		932918100			
9	Manteniment i Serveis - Ecologia Urbana - IN	BCNecologia, l'Agència de Ecologia Urb	C. Escar 1, 3 · 08039 B	info@bcnecol	93 224 0860			
10	Oficina Àrea Verda - BSM	Oficina Àrea Verda - BSM	Calàbria, 66	08015 B	www.areaverc	010		
11	Consorci d'Educació - IMEB	Institut Municipal d'Educació	Plaça d'Espanya, 5. Pl		934023663			
12	Llicències - Ecologia Urbana	BCNecologia, l'Agència de Ecologia Urb	C. Escar 1, 3 · 08039 B	info@bcnecol	93 224 0860			
13	IMI - IdBC - CiesServeis	Institut Municipal d'Informàtica - Insti	Avinguda Diagonal 22		932918100			

Figura 13: Taula que nodreix la fitxa informativa.

El camp “RESULTAT” és la referència vinculada a la taula definitiva de registres que es mostren en pantalla. Quan es clica sobre el botó del registre, mitjançant el sistema explicat en el punt anterior, es va a la taula de resultats i, en aquest cas, s’agafa de la taula el camp “INFO”, i es busca a la taula “INFORMACIÓ” la fila pertinent amb aquest mateix nom. A continuació s’afegeixen tots els camps que contenen informació a la fitxa.

Es pot observar, també, que aquesta taula encara necessita un bon farciment de contingut.

Pel que fa a la fitxa, és senzilla i té aquest aspecte:

Informació

Fitxa amb informació detallada de: IMI - IdBC - Guia ASIA

Nom: Institut Municipal d'Informàtica - Institut de base i cartografia

Adreça: Avinguda Diagonal 220

Telèfon: 932918100

D'acord

Figura 14: Fitxa informativa sobre la referència IMI - IdBC – Guia ASIA





## 6. Algorisme en detall

En aquest punt es detallaran els algorismes més rellevants que permeten que aquesta eina sigui alguna cosa més un cercador.

Com s'ha dit anteriorment, des del primer dia existia una idea molt clara sobre què havia de ser i què no aquest projecte. La màxima prioritat era que fos proactiu, i això va portar a dissenyar una sèrie d'algorismes que, tot aprofitant les cerques realitzades pels usuaris, és a dir, fent ús de les metadades i dissenyant diverses categories per classificar la informació, fessin que l'experiència d'aquest motor de cerca anés més enllà i permetés a l'usuari accedir a informació que potser no hauria sabut trobar o que ni sabia que existia.

Al sistema de reaprofitar la informació amb les metadades de les cerques que fan els propis usuaris se l'ha anomenat "Realimentació", com es veurà a continuació.

### 6.1. La realimentació

#### 6.1.1. En què consisteix

La idea de reciclar la informació que genera cada usuari amb les seves cerques és importantíssima. Prèviament, quan s'han estudiat els tipus de cercadors i motors de cerca existents, s'ha observat la importància que té la recuperació de l'experiència dels usuaris per tal de millorar les futures cerques. Per aquest motiu, s'ha elaborat un algorisme que recupera les metadades generades pels usuaris després de cada cerca realitzada.

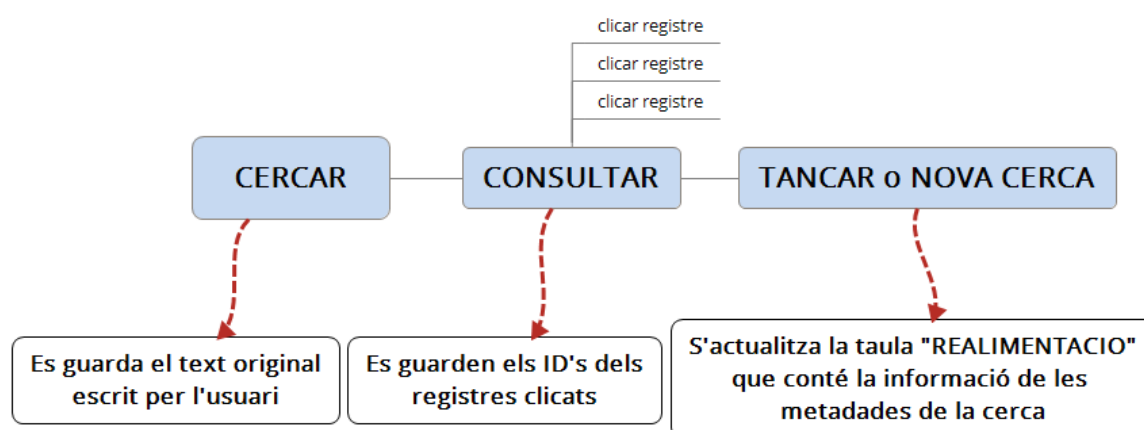


Figura 15: Esquema del funcionament del sistema de realimentació de metadades.

### 6.1.2. L'actualització de les metadades

Aquesta eina ha estat preparada per treballar amb dos tipus de bases de dades diferents. Un tipus és Access, amb la qual es mostren els exemples de taules en aquesta memòria. Tanmateix, on realment es connectarà quan estigui en funcionament és a Oracle. A continuació s'explicarà de quina manera s'actualitzen les metadades que es desen a la taula "REALIMENTACIO".

REALIMENTACIO		
CERCA	INDEXB	VALOR
parcel·les	53	1
licencies	160	1
licencies	13	0,5
licencies	187	0,5
licencies	10	0,5
licencies	43	0,5
licencies	13	1
medi ambient	127	4,4
banderola	214	1
banderola	34	1

Figura 16: Taula "REALIMENTACIO" on es desen les metadades de la informació de les cerques dels usuaris.

Es pot veure que consta de tres camps: "CERCA", "INDEXB" i "VALOR". Els valors dels camps tenen els següents orígens:

- **"CERCA"**: conté la paraula original que l'usuari ha escrit en la seva cerca.

- **"INDEXB"**: conté l'ID del registre o dels registres consultats amb la cerca efectuada i desada en el camp anterior "CERCA".
- **"VALOR"**: emmagatzema un valor que indica la importància del registre amb un ID determinat (camp "INDEXB") un cop s'ha escrit un text concret (camp "CERCA"). El càlcul d'aquest nombre consisteix en sumar, en el cas d'actualitzar; o d'inserir, en el cas d'una nova filera, un número que surt de la següent operació:

$$VALOR = \frac{1}{\text{nombre de registres consultats}}$$

El funcionament de l'algorisme consisteix, en primer lloc, en detectar si s'està en un cas d'actualització d'una fila ja existent o, si altrament, es tracta d'un nou cas i per tant d'un cas d'inserció. Un cop feta la diferenciació, es crida la funció d'actualitzar o la d'inserir segons el cas. L'ordre SQL canvia lleugerament quan es fa a *Oracle* de quan es fa a *Access*. Les ordres són les següents:

- A Oracle:  

```
"UPDATE REALIMENTACIO SET valor =" & val & " WHERE cerca =" & cercaOriginal  
& "' AND indexB =" & indexB
```
- A Access:  

```
"UPDATE REALIMENTACIO SET VALOR = @VALOR WHERE CERCA = @CERCA AND INDEXB =  
@INDEXB"
```

on s'han parametrizat els valors de cada camp de la següent manera:

```
cmd.Parameters.AddWithValue("@VALOR", valorA)  
cmd.Parameters.AddWithValue("@CERCA", cercaOriginal)  
cmd.Parameters.AddWithValue("@INDEXB", indexB)
```

*Fragment de codi 6: Es mostra la manera d'actualitzar la taula "REALIMENTACIO", tant si es tracta d'una taula de dades a Accés com a Oracle.*

La primera conseqüència d'aquesta realimentació és l'alteració de l'ordre en què apareixen els registres en pantalla per a una cerca determinada.

### 6.1.3. L'ordre d'aparició dels registres

L'ordre d'aparició depèn de dues variables. La primera és la que té a veure amb el "VALOR" que s'ha explicat com calcular-lo en el punt anterior, mentre que la segona consisteix en el

nombre total de paraules diferents escrites que tenen coincidències amb les etiquetes d'un registre i s'anomena "Mots Coincidents".

Pel que fa a la primera variable, aquest camp "VALOR" de la taula de dades "REALIMENTACIO" és llegida per l'algorisme quan es troba coincidència. Es llegeix, òbviament, en el cas que existeixi la fila. En les següents línies de codi es pot observar el moment en què, a partir de la coincidència paraula-etiqueta, s'inicia el mecanisme que determina la variable "pes":

```
If LCase(etiquetaNorm).Contains(LCase(paraula)) Then
    Dim pes As Double
    Dim i As Integer = 0

    While i < taulaRealim.Rows.Count
        If LCase(taulaRealim(i)(0)) = cercaNorm And taulaRealim(i)(1) = row(0) Then
            pes = 1 + taulaRealim(i)(2)
            Exit While
        Else
            pes = 1
        End If

        i = i + 1
    End While
```

*Fragment de codi 7: Determinació del pes que té un registre en funció de les dades que es tenen de cerques anteriors guardades a la taula "taulaRealim".*

El que s'observa és que un cop creades les variables "pes" i la que farà de comptador, "i", s'inicia una cerca a la taula on s'ha importat tota la informació de la taula "REALIMENTACIO" en busca del cas on en el camp "CERCA" aparegui la variable "paraula" amb la qual s'ha trobat coincidència i, en el camp "INDEXB", l'ID del registre<sup>[5]</sup> amb el qual s'està avaluant el pes. Si es troba el cas, a la variable pes se li suma el nombre que conté la fila en el camp "VALOR" més 1. En cas contrari, la variable pes valdrà sempre 1.

Pel que fa a la segona variable, s'obté en el moment d'eliminar registres repetits, tal i com s'ha explicat en el punt 5.2.2. *La cerca i l'ajuda del Tesauro i la funció de primitivització*. Com que només es permet guardar un registre per una coincidència per paraula, si estan repetits vol dir que s'han guardat per tenir coincidències amb paraules diferents. D'aquesta manera, quan

---

<sup>[5]</sup> prèviament existeix un "For each" que és una comanda que serveix per dur a terme una sèrie de passos sobre una llista d'objectes, un per un. En aquest cas, els objectes són els registres que han estat seleccionats per ser resultats de la cerca.



s'eliminen els repetits, per a cada eliminació se suma un 1 a la variable "Mots Coincidents" i s'obté una variable indicativa del nombre de paraules escrites per l'usuari que han trobat coincidències en el camp "ETIQUETES" d'un registre.

Finalment, es realitza una multiplicació entre les dues variables i el valor que s'obté serà a partir del qual, en ordre descendent, s'obtindrà l'ordre dels registres mostrats en pantalla.

## 6.2. L'arbre conductor

Un dels significats de voler una eina proactiva és que mai retorni silencis o missatges de l'estil "No hi ha resultats per a la seva cerca. Provi de nou". Amb el cercador per etiquetes, no només s'ha aconseguit tenir un model de motor de cerca i base de dades més simple, sinó que s'ha resolt també aquest assumpte. Gràcies a poder tenir cada registre etiquetat, s'ha pogut confeccionar un arbre que, mitjançant unes classificacions determinades d'etiquetes, permet filtrar les cerques i obtenir resultats sense haver d'escriure absolutament res.

Les classificacions són quatre i són les següents: classificació Open Data Barcelona, Referència de la informació, Temàtica general i Cerques més populars.

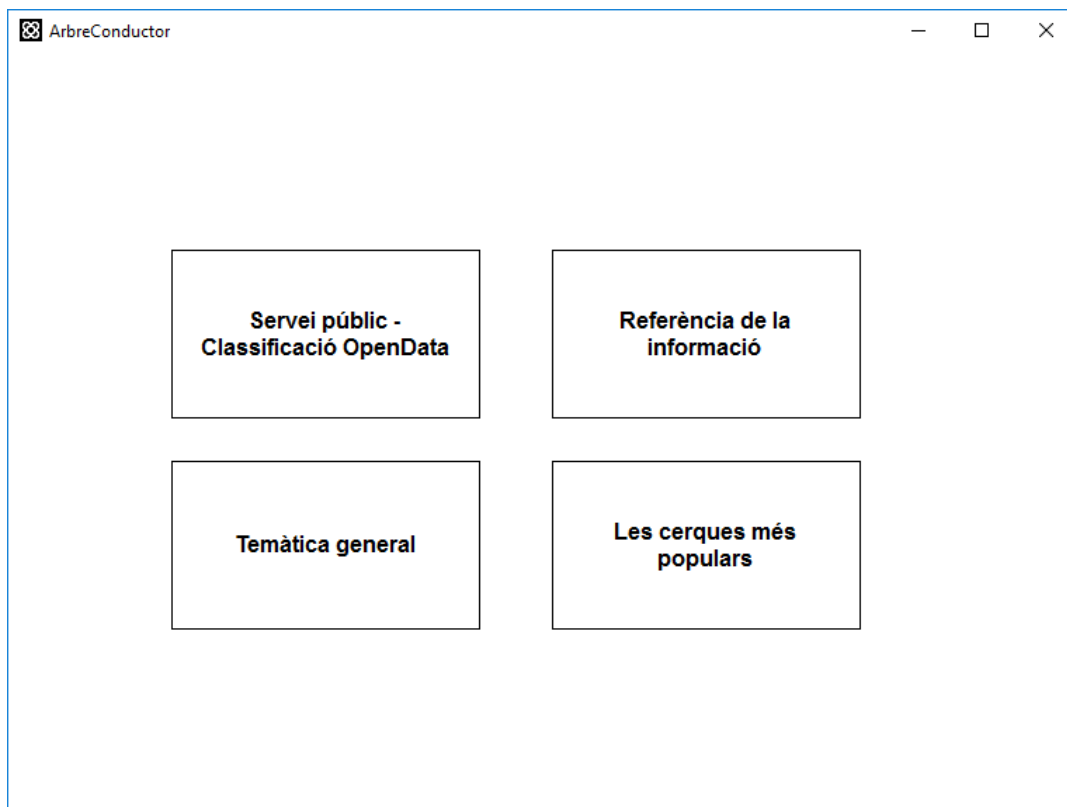


Figura 17: Interfície de la finestra de l'arbre conductor on es veuen les 4 classificacions proposades.

A continuació es descriu què s'hi pot trobar en cada una de les classificacions:

- **Servei públic – Classificació OpenData:** es tracta de classificar les etiquetes segons el tipus de servei públic. La classificació utilitzada és la que utilitza el propi Ajuntament de Barcelona en el seu web d'informació oberta (Open Data) al ciutadà<sup>[6]</sup>.

Si es clica sobre aquesta classificació, apareix la següent pàgina:



<sup>[6]</sup> El web és: <http://opendata.bcn.cat/opendata/ca>



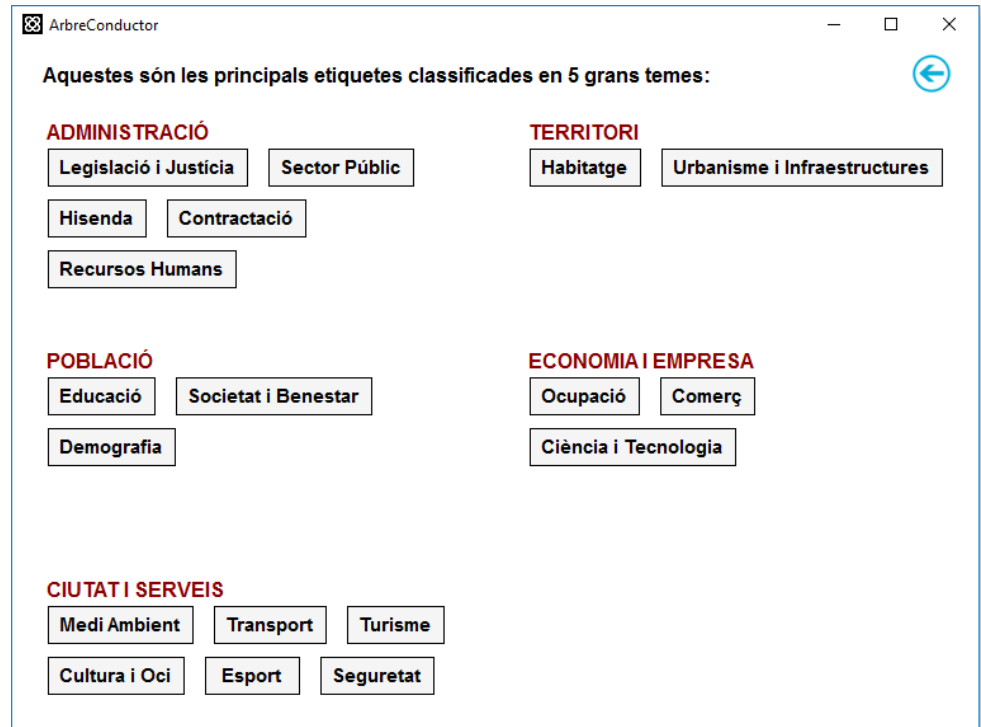


Figura 18: Interfície de la classificació d'etiquetes Open Data Barcelona.

Clicant sobre una de les etiquetes proposades es realitza la cerca i es mostren directament els resultats. En el cas que no interessi aquesta classificació, només cal prémer sobre el botó de la fletxa blava que indica que es torna a la pàgina anterior.

La classificació respon a la pregunta sobre quin tipus d'informació es pot trobar.

- **Referència de la informació:** inclou totes les possibles referències de la informació existents. Quan es clica sobre una de les referències, es fa una cerca simple d'igualtat a la columna "INFO" de la taula *taulaInventari*.

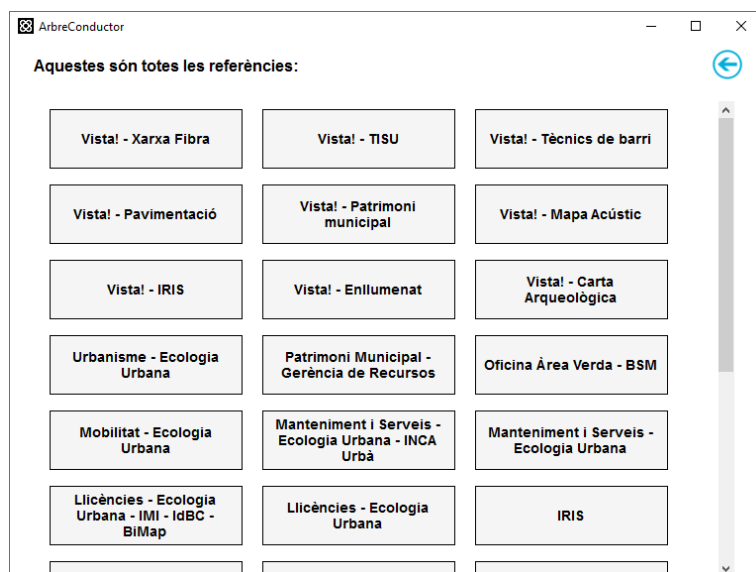


Figura 19: Finestra de la classificació de les referències de la informació.

Aquesta classificació serveix per anar directament a buscar la informació de què disposa un lloc.

- **Temàtica general:** conté tots els temes generals extrets de la taula *taulaInventari*. D'aquesta manera, sempre disposa dels temes actualitzats i òbviament no es mostren temes repetits. En pulsar sobre un dels temes es realitza una cerca estàndard, és a dir, es crida la funció Inicialitzar.





Figura 20: Finestra de la classificació segons la temàtica general.

Aquesta classificació respon a la pregunta sobre quina informació hi ha disponible.

- **Les cerques més populars:** mostra una llista de les cerques més populars basada en el camp "VALOR" de la taula *taulaRealim* que conté les dades importades de la taula d'Access o Oracle "REALIMENTACIO".

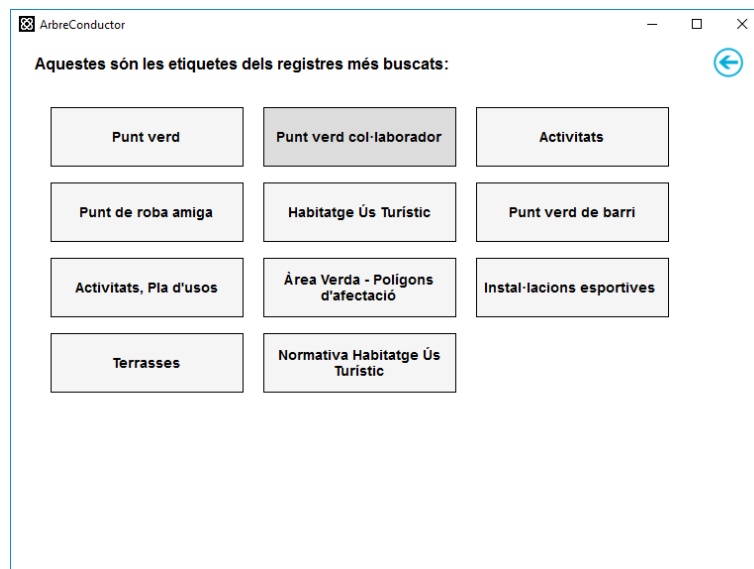


Figura 21: Finestra de la classificació de les cerques més populars.

Clicant a qualsevol de les propostes es realitza una cerca estàndard cridant la funció que inicialitza la cerca com en el cas anterior.

### 6.3. Els suggeriments

Aquesta eina sorgeix de plantejar-se què passa si existeix informació a la qual vol accedir l'usuari però les etiquetes que l'identifiquen no permeten que l'usuari, mitjançant la seva cerca, trobi aquest registre.

La solució no és totalment efectiva, però permet resoldre la situació esmentada en un gran nombre de casos. Consisteix en confiar en què algun usuari, mitjançant la navegació per etiquetes (les quals se suposa que han de ser molt pròximes al que s'està cercant) trobi allò que buscava. En aquest moment es crea un vincle entre el que ha escrit l'usuari al cercador i el que ha consultat, de la mateixa manera que passa sempre, i es guarda a la taula "REALIMENTACIO". El proper cop que un usuari cerqui aquell registre escrivint el mateix text que l'usuari anterior, text amb el qual no havia aconseguit en primera instància obtenir el resultat que volia, es trobarà a la part inferior de la mostra de resultats un panell de suggeriments on hi serà aquest registre en qüestió.

Posant per cas l'exemple següent: un usuari ha escrit "Habitatge" al Cercador, ha obtingut la mostra de resultats pertinent i cap dels resultats li ha fet el pes. Aquest usuari clica sobre l'etiqueta *linkada* "Obres" del registre "Equipaments i habitatge" i obté ara els resultats que contenen aquesta etiqueta. Finalment acaba consultant el registre "Expedients d'edificació i Obres".

Com a conseqüència d'aquesta acció, la propera vegada que aquest usuari o un altre executin la mateixa cerca, obtindran, al final de la mostra de resultats, un panell amb el suggeriment del registre consultat anteriorment "Expedients d'edificació i Obres".

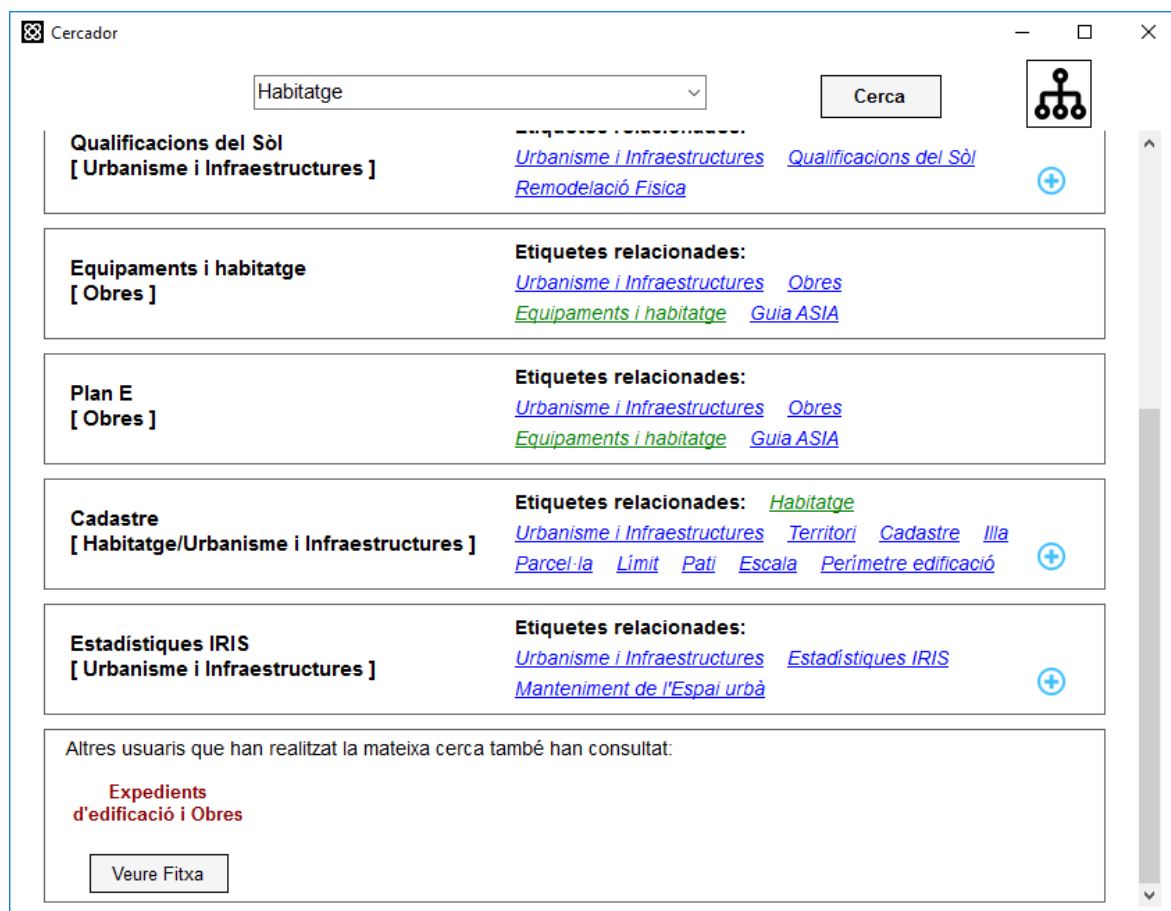


Figura 22: Mostra de resultats de la cerca "Habitatge".

A la figura 22, es pot observar aquest darrer panell amb el suggeriment "Expedients d'edificació i Obres". També es pot observar el registre "Equipaments i habitatge" a través del qual el primer usuari, prement l'etiqueta "Obres", ha aconseguit arribar al registre que ara es mostra com a suggeriment.

Aquesta eina té clarament un pes menor ja que només és un afegit, però encara té recorregut. En aquesta versió no s'ha fet, però existia l'alternativa que, a partir d'un cert nombre de consultes d'aquest suggeriment, s'afegís directament al camp etiquetes del registre aquesta etiqueta que els usuaris escriuen al cercador sense que el registre aparegui com a resultat. En el cas d'exemple anterior, consistiria en afegir l'etiqueta "Habitatge" com a etiqueta del registre "Expedients d'edificació i Obres".



## 7. Parametrització de la informació

La parametrització de la informació consisteix en l'externalització de la definició de variables a un fitxer fàcilment modificable sense haver d'entrar al codi.

### 7.1. Avantatges de la parametrització

La parametrització de la informació ofereix, d'entrada, un enorme avantatge, que és el de poder modificar una variable en un moment sense haver d'entrar al codi i buscar-la d'entre la munió que hi ha definides. I encara és més avantatjós quan resulta que qui necessita modificar la variable no és qui ha fet el codi. Un altre avantatge és que aquest fitxer, com en el cas d'aquest projecte, pot ser un simple fitxer *.ini* que es pot obrir amb el programa *Llibreta* (*Bloc de notas en castellà o notepad en anglès*). L'únic desavantatge és que cal dissenyar un algorisme que llegeixi aquest fitxer i que defineixi les variables en qüestió correctament amb els valors que es proporcionen des del fitxer *.ini*.

### 7.2. Què i com s'ha parametritzat

En el moment que es va proposar la parametrització, la idea era poder alterar les direccions de les taules de dades, les consultes SQL sobre aquestes taules i decidir si el proveïdor de la informació, taula per taula, era *Access* o *Oracle*. Un cop el producte ja es trobava en estat més avançat, es va realitzar un repàs general per veure si hi havia altres variables que també fossin susceptibles de canvis per poder-les també parametritzar. Així doncs, a les anteriors variables de suport a les bases de dades, s'hi han acabat afegint les variables que adopten les direccions de les icones del cercador i les variables que defineixen les propietats de la informació que es mostra en la fitxa informativa, de manera que si es volen canviar camps o afegir-ne (o el que es desitgi fer amb la informació que es vol mostrar), es pugui adaptar fàcilment des del fitxer *.ini* el nom de la nova informació.

A continuació, en el *Fragment de codi 8*, es mostraran alguns exemples d'informació parametritzada:

```
[ConnectionString]
Provider=      Microsoft.ACE.OLEDB.15.0;      Data      Source=      C:\Users\Josep
```

```
Solé\Desktop\bin_Cercador\BBDDcercador.accdb; Persist Security Info=False
Provider= MSDAORA.1; Password= VISTAMAP_P; User ID= VISTAMAP_U; Data Source=
dsv1; Persist Security Info=True
```

```
[ImportTable]
title:taulaAutocompletar
con:1
```

```
select
    Etiquetes
from
    llistatEtiquetes
;
```

```
[ImportTable]
title:taulaInventari
con:2
```

```
select
    *
from
    nouInventari
order by
    IdPropi asc
;
```

*Fragment de codi 8: Es mostren els dos possibles proveïdors (Access i Oracle) i dues de les taules que s'han d'importar amb les consultes SQL que seleccionen què s'importa i l'ordre.*

El codi anterior mostra dos tipus de variables que s'importen des del fitxer *.ini*; en primer lloc, les adreces per poder connectar amb la base de dades, tant a *Access* com a *Oracle* i, en segon lloc, dues de les taules i les seves consultes SQL. La manera de llegir el fitxer per part del codi és anar llegint línia per línia. Quan troba el caràcter “[” sap que ve una nova dada i, a continuació, la llegeix i se'n va a la funció que tracta amb aquesta informació. Una darrera observació sobre el fragment de codi anterior també és el significat del “con:”. El que vol dir és de quin proveïdor, *Access* o *Oracle*, prové la taula que s'ha d'importar.

Pel que fa a les adreces de les icones del Cercador, és ben simple:

```
//Adreces de les icones. Per ordre:
//Cercador; Arbre conductor; SigneMés; SigneMenys; Endarrere
//El número després del nom es refereix als píxels del costat (imatge
quadrada)
```

[Icones]

```
C:\Users\Josep Solé\Desktop\bin_Cercador\Imatges\panot_60.ico  
C:\Users\Josep Solé\Desktop\bin_Cercador\Imatges\tree_view4_37.ico  
C:\Users\Josep Solé\Desktop\bin_Cercador\Imatges\plus_20.ico  
C:\Users\Josep Solé\Desktop\bin_Cercador\Imatges\minus_20.ico  
C:\Users\Josep Solé\Desktop\bin_Cercador\Imatges\back_30.ico
```

*Fragment de codi 9: Es mostren les adreces de les 5 icones que fa servir el Cercador.*

En el fragment de codi anterior s'observa que abans de la definició de les adreces de les icones hi ha unes línies informatives del que ha de venir a continuació. Quan el codi llegeix que els dos primers caràcters d'una línia són "\\" o "--" passa a llegir la següent perquè sap que només es tracta d'una línia informativa per a l'usuari que entra a modificar el fitxer.

Aquesta informació és altament recomanable ja que l'usuari que entra per modificar alguna de les variables que es defineixen en aquest *.ini* necessita saber què és què. En l'exemple anterior, per exemple, si no s'especifica en quin ordre estan definides les adreces de les icones, l'usuari hauria d'anar provant a cegues fins que l'encertés.

Finalment, als fragments de codi anteriors els segueix les definicions de les propietats de la informació que es volen mostrar a la fitxa informativa i que es poden trobar a l'*Annex II*<sup>[7]</sup>.

---

<sup>[7]</sup> L'*Annex II* es pot consultar a la pàgina 98.





## 8. Planificació i cost

### 8.1. Planificació

Amb la finalitat d'enllestir aquest projecte a temps, es va establir un calendari per anar fent el seguiment dels avenços i els estudis de noves millores, amb reunions setmanals. La dedicació al projecte ha estat de quatre hores al dia des de la segona quinzena de març fins al final de la primera quinzena d'agost, de manera que la dedicació setmanal ha estat de vint hores.

A l'inici, abans de començar a escriure codi, la primera tasca que es va dur a terme va ser una documentació prèvia pel projecte en qüestió, la qual va consistir en una pluja d'idees sobre la forma que havia de tenir el cercador, com havia de funcionar el motor de cerca i de quina manera havia de consultar la base de dades, de quina manera es podria resoldre el no retorn de silenci, etc. En definitiva, fer una primera pensada per trobar la manera de complir amb l'abast del projecte. Moltes d'aquestes problemàtiques inicials no s'han resolt fins mesos després, ja que sovint la solució definitiva es veu més clara quan s'ha fet un primer disseny genèric de la funció, interfície o estructura de la base dades.

Poc després de pensar en les primeres solucions, es va començar a estudiar seriosament l'estructura que havia de tenir la base de dades. Com s'ha vist en el punt 4 de la memòria, aquest va ser un tema que es va allargar en el temps més de l'esperat. No va ser fins que es va començar a dissenyar i a fer les primeres proves amb les versions encara més simples del motor de cerca, que es va veure clara quina havia de ser l'organització de l'inventari. Alhora, també es va començar a pensar en la interfície, la qual no va donar, ni molt menys, els mateixos problemes que la organització de l'inventari.

A partir d'una versió prou desenvolupada del motor de cerca, és quan es van començar a elaborar les funcions més complexes que s'havien d'afegir al codi per dotar el cercador d'aquestes funcionalitats afegides que s'havien fixat inicialment.

Els tests de prova-error es van dur a terme des del moment en què l'algorisme ja constava d'una bona base feta i, evidentment, van durar fins al final. De fet, aquesta és una part que mai es deixa de fer mentre el producte segueixi en evolució.

Finalment, es va dur a terme la parametrització d'algunes variables consistent en que la

informació d'aquestes es poguéss modificat des d'un fitxer extern.

La figura de sota és un diagrama de Gantt que representa gràficament la distribució de les diferents tasques al llarg dels mesos en què s'ha realitzat el projecte.

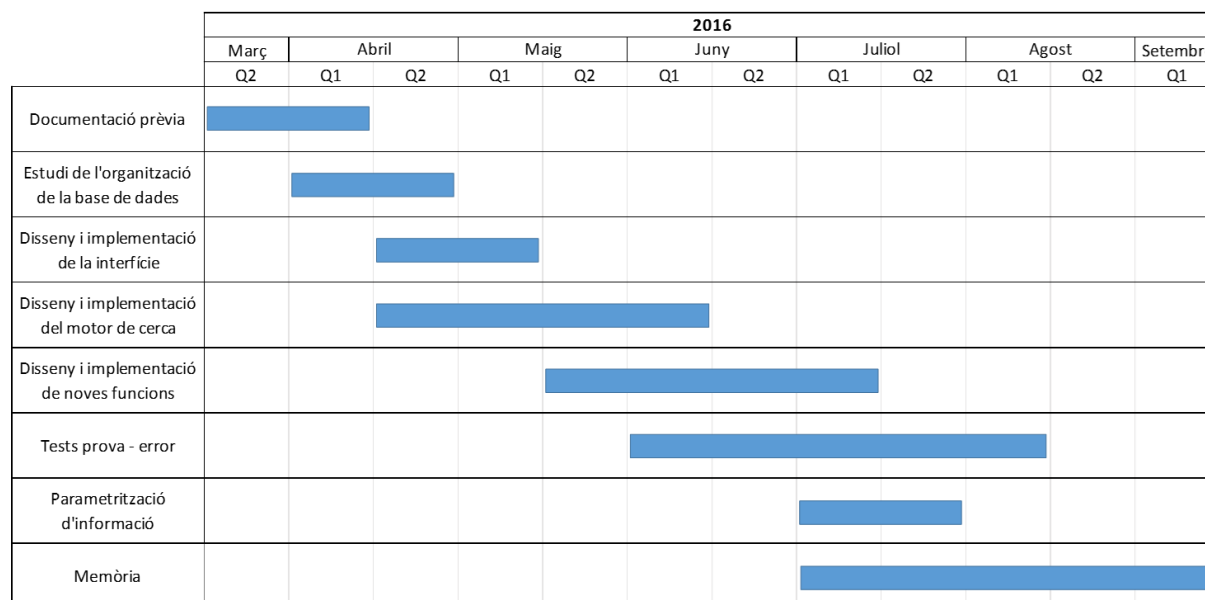


Figura 23: Diagrama de Gantt del projecte.

El pas del temps es representa en l'eix horitzontal i està escalat en quinzenes. D'aquesta manera, cada mes queda dividit en una primera quinzena, Q1, i una segona quinzena, Q2. Les tasques queden representades en l'eix vertical i s'han dividit en un total de 8 que representen, de manera general, totes les tasques realitzades.

## 8.2. Cost

S'ha fet una estimació del possible cost del projecte amb la finalitat de tenir una visió global dels recursos econòmics que ha comportat la realització d'aquest projecte. La gran majoria de les despeses provenen del cost pagat a l'enginyer per tot el temps que hi ha dedicat. Com s'ha explicat en el punt anterior, la dedicació a la feina ha estat de vint hores setmanals. Com que el projecte ha constatat, en total, de 24 setmanes, i es considera que l'enginyer cobra vint euros l'hora, s'ha estimat, en primer lloc, la despesa que ha comportat la realització del projecte per part de l'enginyer. La següent taula resumeix aquest cost:

	<b>Dedicació</b>	<b>Temps total</b>	<b>Taxa</b>	<b>Cost enginyer</b>
<b>Enginyer</b>	20 h/setmana x 24 setmanes	480 h	20 €/h	9600 €
<b>Subtotal</b>				<b>9600 €</b>

Taula 1: Cost del personal.

Alhora, també s'ha considerat la depreciació dels ordinadors portàtils amb què s'ha dut a terme el projecte. D'una banda, des de l'inici i fins la primera quinzena d'agost inclosa, s'ha fet servir un *Toshiba Satellite C50-B-18K*, proporcionat per l'empresa i valorat en uns 400 euros. D'una altra, a partir de la primera quinzena de juliol i fins la finalització del projecte, s'ha fet servir l'equip de l'enginyer, un *Asus A555L* valorat en uns 800 euros, per elaborar la memòria del projecte. S'ha considerat una vida mitja dels equips de 5 anys per al càlcul de la devaluació que es mostra en la *Taula 2*.

	<b>Cost</b>	<b>Vida mitja</b>	<b>Temps d'ús</b>	<b>Amortiment</b>	<b>Cost equipament</b>
<b>Toshiba Satellite C50-B-18K</b>	400 €	5 anys (60 mesos)	5 mesos	8,33 %	33 €
<b>Asus A555L</b>	800 €	5 anys (60 mesos)	2,25 mesos	3,75 %	30 €
<b>Subtotal</b>					<b>63 €</b>

Taula 2: Cost de l'equipament.

Finalment, també s'han considerat les despeses per llicències de software. Afortunadament, no hi ha hagut costos addicionals per llicències ja que els softwares utilitzats han vingut donats per l'ajuntament i pel programari lliure per a estudiants de la UPC. Per tant, el cost per programari ha estat de zero euros.

Per finalitzar l'apartat de costos, s'ha fet una última taula que resumeix tots els costos exposats en les taules anteriors i en fa el càlcul final per obtenir el cost total del projecte.

Concepte	Cost
Enginyer	9600 €
Subtotal 1	9600 €
Amortiment portàtil <i>Toshiba</i>	33 €
Amortiment portàtil <i>Asus</i>	30 €
Subtotal 2	63 €
<b>Total</b>	<b>9663 €</b>

*Taula 3: Taula de costos totals.*

## Conclusions

La solució definitiva del cercador proposada en aquest projecte ha assolit abastament tots els objectius inicials a pesar de semblar d'entrada uns objectius força optimistes. Durant l'elaboració del projecte, s'ha observat que algunes solucions s'han vist forçosament alterades per altres, establint una sèrie de lligams que no s'havien previst inicialment i que quan s'han detectat s'han sabut resoldre correctament. La més destacada ha estat la relació entre l'estructura de la base de dades i la solució del motor de cerca.

El previ estudi de l'estructura de la base de dades ha estat molt més important del que s'havia pensat en el primer moment. Tant és així que les solucions sobre l'organització de l'inventari i el funcionament del motor de cerca han anat de la mà fins que s'ha trobat la solució de cerca per etiquetes. Han estat dos actors que estaven més intrínsecament relacionats del que s'havia previst inicialment i és per aquest motiu que la fase de l'organització de la base de dades ha durat més del previst.

Altrament, també s'ha observat que el cercador dissenyat i implementat té encara molt recorregut. De fet, aquest tipus de producte informàtic sempre està en constant desenvolupament adaptant-se a les necessitats dels seus usuaris. I és que tal i com s'ha dit, aquesta eina haurà de ser utilitzada per més d'onze mil empleats de l'Ajuntament de Barcelona.

Abans, però, caldrà farcir de contingut la base de dades per poder aprofitar al màxim el cercador. Com més informació contingui, més servei farà l'eina dissenyada en aquest projecte. Després de l'actualització de la base de dades, es podrien dur a terme algunes millores que no s'han pogut realitzar en aquesta primera versió. Algunes de les futures possibles millores serien l'addició, dins la fitxa informativa que es mostra a l'usuari en clicar un registre, d'un petit mapa de la ciutat centrat en el punt que correspon a la direcció postal donada. Una altra possible funcionalitat per a la propera actualització podria ser una millora de la interfície, tant a la inicial com a la mostra de resultats. En la primera es pot millorar l'aprofitament de l'espai, ja que actualment queda molt espai en blanc a les parts central i inferior; mentre que en la segona, es podria aguditzar una manera més visual de la mostra de resultats.

Pel que fa a les funcionalitats implementades, la totalitat ha superat satisfactòriament la fase de proves. Els tests prova-error han servit per detectar errades i mancances en diferents

funcions, les quals s'han resolt satisfactòriament.

En conclusió, aquest projecte ha estat capaç d'abordar els objectius que s'havien marcat definits per les hipotètiques necessitats estudiades prèviament que haurien de tenir els usuaris d'aquest cercador. Els costos han estat realment baixos i la planificació, en general, s'ha complert. Això no obstant, i com que es tracta d'un software que ha d'estar en millora constant, fàcilment es podrien tornar a marcar uns nous objectius per a la propera versió d'aquest producte, alguns dels quals s'acaben de comentar.

Amb la consciència que aquest producte s'havia de dissenyar perquè fos entenedor i, sobretot, després d'aquesta primera versió, tingués continuïtat, sempre s'ha pensat la manera de fer més idònia per complir amb aquest objectiu. Alguns exemples en són l'extens estudi de la manera de cercar a la base de dades que va acabar amb el model de cerca per etiquetes, la parametrització de variables que permet accedir a modificar les variables que s'han considerat oportunes a través d'un fitxer extern o la classificació per mòduls del codi on els diferents algorismes queden millor organitzats per facilitar la comprensió del futur enginyer que hagi d'accedir al codi.

## Agraïments

A l'Isaac Aparicio, responsable d'informació de base a l'Institut Municipal d'Informàtica de l'Ajuntament de Barcelona i tutor de les pràctiques curriculars realitzades a l'IMI, per ser el cercador una idea original seva i per la confiança dipositada i la llibertat atorgada en la realització d'aquest projecte.





## Bibliografia

### Referències bibliogràfiques

- [1] HUMAN LEVEL COMMUNICATIONS. *¿Cómo funciona un buscador como Google? – Parte I: Indexación.* Ramón Saquete, 2014. [<http://www.humanlevel.com/articulos/desarrollo-web/como-functiona-un-buscador-como-google-indexacion.html>, 22 de març de 2016].
- [2] STACK OVERFLOW. *How to insert a value that contains an apostrophe (single quote)?*. Peter Mortensen, 2009. [<http://stackoverflow.com/questions/1912095/how-to-insert-a-value-that-contains-an-apostrophe-single-quote>, 26 d'abril de 2016].
- [3] EDITORIAL LUMEN. *Cercador*. Red Net Argentina, 2016. [<http://www.edlumen.net>, 4 de maig de 2016].
- [4] SNOWBALL. *Spanish stemming algorithm*. Dr Martin Porter, 2002. [<http://snowball.tartarus.org/algorithms/spanish/stemmer.html>, 8 d'abril de 2016].
- [5] SERVEI DE DADES OBERTES DE L'AJUNTAMENT DE BARCELONA. *Catàleg de dades obertes*. Barcelona, 2016. [<http://opendata.bcn.cat/opendata/ca>, 30 de maig de 2016].

### Bibliografia complementària

Fòrums de Visual Basic per fer consultes sobre el codi:

- [6] LA WEB DEL PROGRAMADOR. *Fòrum de Visual Basic*. Interactive Programmers Community, 2016. [<http://www.lawebdelprogramador.com/foros/Visual-Basic/index1.html>, diverses consultes entre març i agost de 2016].
- [7] MICROSOFT. *Fòrum de Visual Basic*. Microsoft, 2016. [<https://social.msdn.microsoft.com/Forums/vstudio/en-US/home?forum=vbgeneral>, diverses consultes entre març i agost de 2016].

- [8] CANAL VISUAL BASIC .NET, Fòrum de Visual Basic. canalvisualbasic.net, 2016. [<http://www.canalvisualbasic.net/foro/visual-basic-6-0>], diverses consultes entre març i agost de 2016].

## Annex I: Codi Visual Basic

El codi es presenta en 3 parts. La primera és la central, que conté la gran majoria de tot el codi i està dividida en mòduls, la segona fa referència a l'arbre conductor i la tercera a la fitxa informativa.

### Part 1: Codi central

```
Imports System.Data
Imports System.Data.OleDb

Public Class Cercador
    '***** DEFINICIÓ DE VARIABLES GLOBALS, TAULES GLOBALS, ETC *****
    '***** Llista de rutes de BBDD i adaptador.
    Public llistaConnectionString As New List(Of String)
    Dim connDB As New OleDbConnection

    '*** Taules que es creen a l'inici.
    Public taulaAutocom As New DataTable
    Public taulaInventari As New DataTable
    Public taulaInformacio As New DataTable

    '*** Taules que es creen sobre la marxa.
    Public taulaRealim As New DataTable
    Public consultaSQL_taulaRealim As String
    Public connection4 As String
    Public consultaSQL_tesaurus As String
    Public connection5 As String
    Public consultaSQL_OpenData As String
    Public connection6 As String

    '*** Resta de variables globals.
    Public taula As New DataTable
    Public cercaNorm As String
    Public cercaOriginal As String
    Public cercaMod As String
    Public w_mod(Len(nParaules)) As String
    Public dir_exe As String
    Public clics As Integer = 0
    Public nParaules As New Integer
    Public valor As New Double
    Public pos_scroll As Integer
    Public Titol_fitxaInfo As String
    Public propietats_info As New List(Of String)
    Private ORATrue_ACCfalse As Boolean
    Private botoEnrere As New PictureBox

    '*** Llistes d'elements que es creen durant el transcurs de la cerca.
    Public llistaClics As New List(Of Integer)
    Public llistaPanells As New List(Of Panel)
    Public llistaCaixesEtiquetes As New List(Of FlowLayoutPanel)
    Public llistaBotonsPlus As New List(Of PictureBox)
```

```

Public llistaBotonsMinus As New List(Of PictureBox)
Public llistaCerques As New List(Of String)

'*** Icones del cercador (maximitzar, minimitzar i enrere).
Public imatge_plus As Image
Public imatge_minus As Image
Public imatge_back As Image

'*** Per les diferents taules que s'importen d'Access.
Private da1 As OleDbDataAdapter
Private da2 As OleDbDataAdapter
Private da3 As OleDbDataAdapter
Private da4 As OleDbDataAdapter
Private da5 As OleDbDataAdapter

'*****
'*****

'*****
'***** EL QUE CARREGA QUAN S'INICIA L'EXECUTABLE *****
'*****

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    '* Adreça de la ubicació de l'executable.
    dir_exe = Application.StartupPath()

    LoadIni()

    ComboBox1.DropDownStyle = ComboBoxStyle.DropDown
    ComboBox1.Text = ""
    Panel1.SendToBack()

    Dim ToolTip1 As New ToolTip()
    With ToolTip1
        .AutoPopDelay = 5000
        .InitialDelay = 500
        .ReshowDelay = 300
        .ForeColor() = Color.White
        .BackColor() = Color.Black
        .ShowAlways = True
        .SetToolTip(BotoArbre, "Menú de l'arbre conductor")
    End With

    Me.Controls.Add(botoEnrere)
    With botoEnrere
        .Image = imatge_back
        .Location = New System.Drawing.Point(30, ComboBox1.Location.Y)
        .AutoSize = True
        .Visible = False
    End With
    AddHandler botoEnrere.Click, AddressOf ClickEvent_CercaAnterior

End Sub

'*** Funció que crida l'INI que conté informació sobre rutes de les BBDD, rutes
'***_d'imatges de les icones i informació sobre els continguts que es mostraran a la
'***_fitxa del registre.
Private Sub LoadIni()

```

```

Me.SetScrollState(ScrollStateFullDrag, False)
Dim liniaINI As String

'* Per tal que es mostrin els accents es realitzarà la següent comanda:
Dim oRead As System.IO.StreamReader = New System.IO.StreamReader(dir_exe &
"\Cercador.ini", System.Text.Encoding.Default)

While oRead.Peek <> -1
    '* Si el primer caràcter de la línia es un "[" --> La línia és
    '* _capçalera de secció.
    If Chr(oRead.Peek) = "[" Then
        liniaINI = oRead.ReadLine()

        '* Es passa a minúscules per evitar problemes de confusions entre
        '* _majúscules i minúscules.
        liniaINI = LCase(liniaINI)

        Select Case liniaINI
            Case "[connectionstring]"
                def_ConnectionString(oRead)

            Case "[importtable]"
                def_ImportTable(oRead)

            Case "[titolfitxa]"
                def_TitolFitxa(oRead)

            Case "[propietatsinfo]"
                def_Info(oRead)

            Case "[icones]"
                def_icons(oRead)

            Case Else

        End Select
    Else
        oRead.ReadLine()
    End If

End While
oRead.Close()

End Sub

'*** Funció que llegeix el/s ConnectionString.
Private Sub def_ConnectionString(ByRef lector As System.IO.StreamReader)
    Dim liniaINI As String

    Try
        '* Si el següent caràcter que es llegeix és un EOF (End Of File) se
        '* _surts del procediment.
        '* Si el següent caràcter que es llegeix és un "[" se surt del
        '* _procediment.
        While ((lector.Peek <> -1) And (lector.Peek <> 91))
            liniaINI = lector.ReadLine

            If liniaINI = "" Then
                Exit While
            End If
        End While
    Catch
    End Try
End Sub

```

```

ElseIf (liniaINI.Substring(0, 2) <> "--") And (liniaINI.Substring(0,
2) <> "//") Then
    '* S'afegeixen tots elsConnectionString a la
    '* _llistaConnectionString.
    llistaConnectionString.Add(liniaINI)

End If

End While

Catch ex As Exception
    MessageBox.Show("Error en tractar la cadena de connexió")
End Try
End Sub

'*** Funció que llegeix i importa les taules de la/les BBDD.
Private Sub def_ImportTable(ByRef lector As System.IO.StreamReader)
    Dim liniaINI As String
    Dim connection As String

    '* Si el següent caràcter que es llegeix és un EOF (End Of File) se surt
    '* del procediment.
    '* Si el següent caràcter que es llegeix és un "[" se surt del
    '* procediment.
    While ((lector.Peek <> -1) And (lector.Peek <> 91))
        liniaINI = lector.ReadLine

        If liniaINI = "" Then '* Si la línia és un blanc, no es fa res.

ElseIf (liniaINI.Substring(0, 2) <> "--") And (liniaINI.Substring(0, 2)
<> "//") Then
    '* Es defineix un vector on guardar el nom de la propietat i el
    '* seu valor.
    Dim prop_val() As String
    Dim nom_taula As String
    Dim con As Integer

    prop_val = liniaINI.Split(":")

    '* Per evitar problemes de majúscules y minúscules, es passa el
    '* nom de la taula a minúscules.
    nom_taula = LCase(prop_val(1))

    '* Aquí es llegeix a quin ConnectionString fa referència.
    liniaINI = lector.ReadLine
    prop_val = liniaINI.Split(":")
    con = CInt(prop_val(1)) - 1
    connection = llistaConnectionString(con)

    '* Aquí comença la consulta SQL.
    liniaINI = lector.ReadLine
    Dim consulta_sql As String
    consulta_sql = " "

    While (LCase(liniaINI) <> ";")
        '* Si la línia és un blanc, no es fa res.
        If liniaINI = "" Then

```

```

        ElseIf (liniaINI.Substring(0, 2) <> "--") And
(liniaINI.Substring(0, 2) <> "//") Then
            consulta_sql = consulta_sql & liniaINI.Trim() & " "

        End If

        liniaINI = lector.ReadLine()
    End While

    Select Case nom_taula
        Case "taulaautocompletar"
            '* Emplenar un Datatable amb la info del llistat
            '* d'etiquetes.
            da1 = New OleDbDataAdapter(consulta_sql, connection)
            da1.Fill(taulaAutocom)

        Case "taulainventari"
            '* Emplenar un Datatable amb la info d'inventari.
            da2 = New OleDbDataAdapter(consulta_sql, connection)
            da2.Fill(taulaInventari)

        Case "taulainformacio"
            '* Emplenar un Datatable amb la info de la procedència de
            '* la informació.
            da3 = New OleDbDataAdapter(consulta_sql, connection)
            da3.Fill(taulaInformacio)

        Case "taularealimentacio"
            consultaSQL_taulaRealim = consulta_sql
            connection4 = connection
            connDB.ConnectionString = connection
            If connection.Contains("MSDAOA") Then
                ORAtrue_ACCfalse = True

            ElseIf connection.Contains("Microsoft") Then
                ORAtrue_ACCfalse = False

            Else
                ORAtrue_ACCfalse = True

            End If

        Case "taulatesaurus"
            consultaSQL_tesaurus = consulta_sql
            connection5 = connection

        Case "taulaopendata"
            consultaSQL_OpenData = consulta_sql
            connection6 = connection

    End Select

End If

End While

End Sub

'*** Funció que llegeix i importa el títol de la fitxa d'informació del registre.

```

```

Private Sub def_TitolFitxa(ByRef lector As System.IO.StreamReader)
    Dim liniaINI As String

    Try
        '* Si el següent caràcter que es llegeix és un EOF (End Of File) se
        '* _surt del procediment.
        '* Si el següent caràcter que es llegeix és un "[" se surt del
        '* _procediment.
        While ((lector.Peek <> -1) And (lector.Peek <> 91))
            liniaINI = lector.ReadLine

            If liniaINI = "" Then
                Exit While

            ElseIf (liniaINI.Substring(0, 2) <> "--") And (liniaINI.Substring(0,
2) <> "//") Then
                '* S'afegirà el títol de la fitxa.
                Titol_fitxaInfo = liniaINI

            End If

        End While

        Catch ex As Exception
            MessageBox.Show("Error en tractar la cadena de connexió.")
        End Try
    End Sub

    '*** Funció que llegeix i importa les propietats de la info que estan
    '***_definides a la BBDD.
    Private Sub def_Info(ByRef lector As System.IO.StreamReader)
        Dim liniaINI As String

        Try
            '* Si el següent caràcter que es llegeix és un EOF (End Of File) se
            '* _surt del procediment.
            '* Si el següent caràcter que es llegeix és un "[" se surt del
            '* _procediment.
            While ((lector.Peek <> -1) And (lector.Peek <> 91))
                liniaINI = lector.ReadLine

                If liniaINI = "" Then
                    Exit While

                ElseIf (liniaINI.Substring(0, 2) <> "--") And (liniaINI.Substring(0,
2) <> "//") Then
                    '* S'afegiran totes les propietats a la llista
                    '*_propietats_info.
                    propietats_info.Add(liniaINI)

                End If

            End While

            Catch ex As Exception
                MessageBox.Show("Error en tractar la cadena de connexió.")
            End Try
        End Sub

```



```

    *** Funció que llegeix i importa les icones que s'utilitzen en el Cercador.
    Private Sub def_icons(ByRef lector As System.IO.StreamReader)
        Dim liniaINI As String

        Try
            '* Si el següent caràcter que es llegeix és un EOF (End Of File) se
            '*_surt del procediment.
            '* Si el següent caràcter que es llegeix és un "[" se surt del
            '*_procediment.
            While ((lector.Peek <> -1) And (lector.Peek <> 91))
                liniaINI = lector.ReadLine

                If liniaINI = "" Then
                    Exit While

                ElseIf (liniaINI.Substring(0, 2) <> "--") And (liniaINI.Substring(0,
2) <> "//") Then
                    Icon = New System.Drawing.Icon(liniaINI)

                    liniaINI = lector.ReadLine
                    BotoArbre.Image = Image.FromFile(liniaINI)

                    liniaINI = lector.ReadLine
                    imatge_plus = Image.FromFile(liniaINI)

                    liniaINI = lector.ReadLine
                    imatge_minus = Image.FromFile(liniaINI)

                    liniaINI = lector.ReadLine
                    imatge_back = Image.FromFile(liniaINI)
                End If

            End While

        Catch ex As Exception
            MessageBox.Show("Error en tractar les adreces de les icones.")
        End Try
    End Sub

    *****
    *****

    *****
    ***** ESDEVENIMENTS SOBRE EL COMBOBOX I LA INICIALIZACIÓ DE LA CERCA *****
    *****

    *** Esdeveniment quan canvia el text del ComboBox.
    Private Sub ComboBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.TextChanged
        Dim text As String
        text = TreureApostrof(ComboBox1.Text)

        If ComboBox1.SelectedItem Is Nothing Then
            ComboBox1.Items.Clear()
        Else
            Exit Sub
        End If

        If ComboBox1.Text = "" Then
            ComboBox1.DropDownStyle = ComboBoxStyle.DropDown

```

```

        Exit Sub
    Else
        ComboBox1.DropDownStyle = ComboBoxStyle.Simple
        ComboBox1.SelectionStart = ComboBox1.Text.Length
    End If

    Dim etiq_like() As DataRow
    etiq_like = taulaAutocom.Select("Etiquetes LIKE '%" & text & "%'")

    For Each dr As DataRow In etiq_like
        ComboBox1.Items.Add(dr("Etiquetes").ToString)
    Next
End Sub

'*** Esdeveniment quan es prem la tecla Intro en el ComboBox.
Private Sub ComboBox1_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles ComboBox1.KeyPress
    If e.KeyChar = Chr(13) Then
        If ComboBox1.Text = "" Or ComboBox1.Text = " " Or ComboBox1.Text = "  "
Then Exit Sub

        cercaOriginal = LCase(NormalitzarString(ComboBox1.Text))
        Inicialitzar(ComboBox1.Text)
    End If

    If clics > 0 Then
        CalculClics()
    End If
End Sub

'*** Esdeveniment quan es prem el botó Cerca.
Private Sub BotoCerca_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BotoCerca.Click
    If ComboBox1.Text = "" Or ComboBox1.Text = " " Or ComboBox1.Text = "  " Then
Exit Sub

        cercaOriginal = LCase(NormalitzarString(ComboBox1.Text))
        Inicialitzar(ComboBox1.Text)

        If clics > 0 Then
            CalculClics()
        End If
    End Sub

'*** Esdeveniment en entrar al Combobox.
Private Sub ComboBox1_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.Enter
    ComboBox1.DropDownStyle = ComboBoxStyle.Simple
    ComboBox1.BringToFront()
End Sub

'*** Esdeveniment en sortir del Combobox.
Private Sub ComboBox1_Leave(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.Leave
    ComboBox1.DropDownStyle = ComboBoxStyle.DropDown
End Sub

'*** Esdeveniment en clicar sobre el Panel1.

```

```

Private Sub Panel1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Panel1.Click
    ComboBox1.DropDownStyle = ComboBoxStyle.DropDown
    Panel1.Focus()
End Sub

'*** Esdeveniment en clicar sobre el Form.
Private Sub Me_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Me.Click
    ComboBox1.DropDownStyle = ComboBoxStyle.DropDown
    Panel1.Focus()
End Sub

'*****
'*****
'***** EXECUCIÓ DE LA CERCA *****
'*****
'*** Funció que posa en marxa la cerca.
Private Sub Inicialitzar(ByVal cerca)
    nParaules = 1
    Actualitzar_taulaRealim()

    cercaNorm = NormalitzarString(cerca)
    cercaNorm = TreureApostrof(cercaNorm)
    cercaMod = cercaNorm

    If llistaCerques.Count = 0 Then
        llistaCerques.Add(cerca)

    Else
        If cerca = llistaCerques.ElementAt(llistaCerques.Count - 1) Then

        Else
            llistaCerques.Add(cerca)

        End If
    End If

    Me.Cercar()
    Me.AfegirRegistres()

    ComboBox1.Text = cerca

End Sub

'*** Funció que defineix les taules necessàries per realitzar la cerca, crida
'***_la cerca i crida l'ordenació dels resultats.
Private Sub Cercar()
    Try
        taula.Columns.Add("Índex", GetType(Integer))
        taula.Columns.Add("Tema", GetType(String))
        taula.Columns.Add("Subtema", GetType(String))
        taula.Columns.Add("Etiquetes", GetType(String))
        taula.Columns.Add("Informació", GetType(String))
        taula.Columns.Add("Valor", GetType(Double))
    Catch ex As Exception
        If taula.Rows.Count > 0 Then

```

```

        Panel1.Controls.Clear()
        taula.Rows.Clear()
        llistaPanells.Clear()
        llistaCaixesEtiquetes.Clear()
        llistaBotonsPlus.Clear()
        llistaBotonsMinus.Clear()
    End If
End Try

Dim taulaCoincidencies As New DataTable
Try
    taulaCoincidencies.Columns.Add("Índex", GetType(Integer))
    taulaCoincidencies.Columns.Add("Tema", GetType(String))
    taulaCoincidencies.Columns.Add("Subtema", GetType(String))
    taulaCoincidencies.Columns.Add("Etiquetes", GetType(String))
    taulaCoincidencies.Columns.Add("Informació", GetType(String))
    taulaCoincidencies.Columns.Add("Pes", GetType(Double))
    taulaCoincidencies.Columns.Add("Mots Coincidents", GetType(Integer))
Catch ex As Exception
    taulaCoincidencies.Rows.Clear()
End Try

CercarInventari(taulaCoincidencies)
OrdenarRegistres(taulaCoincidencies)

End Sub

'*** Funció que fa la cerca i selecciona els registres que continguin la o
'***_les paraules cercades en les seves etiquetes.
Private Sub CercarInventari(ByRef taulaCoincidencies As DataTable)
    Dim i As Integer
    Dim j As Integer
    Dim etiquetaNorm As String

    If cercaNorm = " " Or cercaNorm = "" Or cercaNorm.Length = 0 Then Exit Sub

    '* Buscar a la taulaInventari, en la columna etiquetes.
    For Each row As DataRow In taulaInventari.Rows
        Try
            etiquetaNorm = NormalitzarString(row(3))
        Catch ex As Exception
            etiquetaNorm = ""
        End Try

        If LCase(" " & etiquetaNorm).Contains(LCase(" " & cercaNorm)) Then
            Dim pes As Double

            i = 0
            While i < taulaRealim.Rows.Count

                If LCase(taulaRealim(i)(0)) = LCase(cercaNorm) And
taulaRealim(i)(1) = row(0) Then
                    pes = 1 + taulaRealim(i)(2)
                    Exit While
                Else
                    pes = 1
                End If
            End While
        End If
    End For
End Sub

```

```

        i = i + 1
    End While

    taulaCoincidencies.Rows.Add(New Object() {row(0), row(1), row(2),
row(3), row(4), pes, 1})

    Else
    End If
Next

'* Es prepara l'algorisme per realitzar el Tesaurus.
Dim filesOK, troballesPerParaula As Integer
Dim equiv As String
equiv = NormalitzarString(Tesaurus(cercaNorm))

'*****
'*** Si el Tesaurus ha retornat una nova paraula, es fa la cerca amb la
'***_nova paraula. Sinó, no cal.
'*****
If LCase(cercaNorm) = LCase(equiv) Then
Else
    CercarEtiqueta(equiv, filesOK, troballesPerParaula, taulaCoincidencies)

    '* Si s'ha afegit un registre i aquest ja hi era, s'elimina.
    '* Condició per saber si s'han afegit files.
    If taulaCoincidencies.Rows.Count > filesOK Then
        cercaMod = equiv
        i = 0
        While i < taulaCoincidencies.Rows.Count

            j = i + 1
            While j < taulaCoincidencies.Rows.Count

                If taulaCoincidencies(i)(0) = taulaCoincidencies(j)(0) Then
                    taulaCoincidencies.Rows.RemoveAt(j)

                Else
                    j = j + 1
                End If
            End While

            i = i + 1
        End While
    End If
End If

'*****
'*** Si no s'ha trobat una etiqueta que contingui el que s'ha escrit al
'***_cercador, es busca paraula per paraula.
'*****
If taulaCoincidencies.Rows.Count = 0 Then
    '* Es separen les paraules sense preposicions ni conjuncions i es
    '*_compten.
    cercaNorm = TreurePreposicions(cercaNorm)
    cercaNorm = TreureEspaisEnBlanc(cercaNorm)

    nParaules = 1
    For Each caracter As String In cercaNorm

```

```

        '* Per cada espai en blanc, se suma una paraula.
    If character = " " Then
        nParaules = nParaules + 1

    End If
Next

'* Es crea una llista amb les respectives paraules separades.
Dim w(Len(nParaules)) As String
w = Split(cercaNorm, " ")
w_mod = w

j = 0
While j < nParaules
    If nParaules > 1 Then
        troballesPerParaula = 0

        CercarEtiqueta(w(j), filesOK, troballesPerParaula,
taulaCoincidencies)

        equiv = NormalitzarString(Tesaurus(w(j)))
        '* Si el Tesaurus ha retornat una nova paraula, es fa la
        '* _cerca. Sinó, no cal.
        If LCase(w(j)) = LCase(equiv) Then

            Else
                CercarEtiqueta(equiv, filesOK, troballesPerParaula,
taulaCoincidencies)
                w_mod(j) = equiv
            End If
        End If
        '*****
        '*** Si no s'han trobat resultats amb el Tesaurus, es fa la
        '*** _primitiva de la paraula.
        '*****
        If troballesPerParaula = 0 Then
            Dim primitiva As String
            primitiva = FerArrel(w(j))
            CercarEtiqueta(primitiva, filesOK, troballesPerParaula,
taulaCoincidencies)

            If troballesPerParaula > 0 Then
                w_mod(j) = primitiva
                cercaMod = primitiva
            End If
        End If

        j = j + 1
    End While

    '*****
    '* Cal eliminar els registres repetits i, per cada eliminació, sumar
    '* _una unitat a la columna 6 "mots coindidents".
    '*****
    If taulaCoincidencies.Rows.Count > 1 Then
        i = 0 '* Fila "i" de referència per comparar-la amb les de sota.
        j = 1 '* Comptador de files per comparar-se amb la fila "i"
        Dim k As Integer = 1 '* Comptador de files iguals.

```

```

        While i < taulaCoincidencies.Rows.Count
            j = i + 1
            While j < taulaCoincidencies.Rows.Count
                If taulaCoincidencies(i)(0) = taulaCoincidencies(j)(0) Then
                    taulaCoincidencies.Rows.RemoveAt(j)
                    k = k + 1
                    taulaCoincidencies(i)(6) = k
                Else
                    j = j + 1
                End If
            End While

            i = i + 1
            k = 1
        End While
    Else
        End If
    End If
End Sub

''' Cerca si existeixen coincidències per la paraula entrant i retorna el
'''_nombre de files que es portaven, un comptador de coincidències per la
'''_paraula entrant i la taula actualitzada amb els nous registres.
Private Sub CercarEtiqueta(ByVal paraula As String, ByRef filesOK As Integer,
ByRef n As Integer, ByRef taulaCoincid As DataTable)
    If paraula = " " Or paraula = "" Then Exit Sub

    filesOK = taulaCoincid.Rows.Count
    n = 0 '* Comptador de troballes per paraula.
    Dim etiquetaNorm As String

    For Each row As DataRow In taulaInventari.Rows
        Try
            etiquetaNorm = NormalitzarString(row(3))
        Catch ex As Exception
            etiquetaNorm = ""
        End Try

        '* Buscar a la taulaInventari, a la columna etiquetes.
        If LCase(etiquetaNorm).Contains(LCase(paraula)) Then
            Dim pes As Double
            Dim i As Integer = 0

            While i < taulaRealim.Rows.Count
                If LCase(taulaRealim(i)(0)) = cercaNorm And taulaRealim(i)(1) =
row(0) Then
                    pes = 1 + taulaRealim(i)(2)
                    Exit While
                Else
                    pes = 1
                End If

                i = i + 1
            End While

            taulaCoincid.Rows.Add(New Object() {row(0), row(1), row(2), row(3),
row(4), pes, 1})

```

```

        n = n + 1
    Else
    End If
Next
End Sub

'*** Copiar les files a la taula en ordre DESC del producte entre
'***_MotsCoincidents pel Pes.
Private Sub OrdenarRegistres(ByVal taulaCoinc As DataTable)
    Dim j As Integer = nParaules
    Dim i As Integer
    Dim taulaDesordenada As New DataTable

    Try
        taulaDesordenada.Columns.Add("Índex", GetType(Integer))
        taulaDesordenada.Columns.Add("Tema", GetType(String))
        taulaDesordenada.Columns.Add("Subtema", GetType(String))
        taulaDesordenada.Columns.Add("Etiquetes", GetType(String))
        taulaDesordenada.Columns.Add("Informació", GetType(String))
        taulaDesordenada.Columns.Add("Valor", GetType(Double))
    Catch ex As Exception
        taulaDesordenada.Rows.Clear()
    End Try

    While j > 0
        i = 0

        '* Condició: si ja hi ha almenys 5 resultats amb més paraules
        '*_coincidents que els següents a afegir, FI!
        If taulaDesordenada.Rows.Count < 5 Then
            While i < taulaCoinc.Rows.Count

                If taulaCoinc(i)(6) = j Then
                    taulaDesordenada.Rows.Add(New Object() {taulaCoinc(i)(0),
taulaCoinc(i)(1), taulaCoinc(i)(2), taulaCoinc(i)(3), taulaCoinc(i)(4),
Cdbl(taulaCoinc(i)(5)) * Cdbl(taulaCoinc(i)(6))})
                Else
                End If

                i = i + 1
            End While
        End If

        j = j - 1
    End While

    '* Es crea un Dataview a partir de la taula amb els resultats de la
    '*_cerca.
    Dim view As New DataView(taulaDesordenada)
    view.Sort = "Valor DESC" '* S'ordena el Dataview per valor descendent.
    taula = view.ToTable() '* Es crea la taula definitiva ordenada.
End Sub

'*** Funció que crea la disposició final del registres per ser visualitzats.
Private Sub AfegirRegistres()
    Dim i As Integer = 0
    Dim y As String = 10
    Dim llistaEtiquetes As New List(Of String)
    Dim etiquetes() As String

```



```

If taula.Rows.Count = 0 Then
    With ArboreConductor
        .Show()
        .Focus()
        .Label1.Visible = False
        .Label2.Visible = True
        .Label2.SendToBack()
    End With

Else
    Panel1.BringToFront()
    Panel1.Visible = True
    ComboBox1.DropDownStyle = ComboBoxStyle.DropDown

    While i < taula.Rows.Count
        Dim panell As New Panel
        Panel1.Controls.Add(panell)
        With panell
            .Name = i.ToString & "-Panell"
            .Location = New System.Drawing.Point(10, y)
            .Width = Panel1.Width - 50
            .AutoSizeMode = AutoSizeMode.GrowOnly
            .Height = 80
            .BorderStyle = BorderStyle.FixedSingle
            .Show()
        End With

        Dim Subtema As String = taula(i)(2).ToString
        Dim Tema As String = taula(i)(1).ToString

        Dim botoInfo As New Button
        panell.Controls.Add(botoInfo)
        With botoInfo
            .Name = i.ToString & "-botoInfo"
            .Location = New System.Drawing.Point(10, 5)
            .Width = 310
            .Height = 65
            .FlatStyle = FlatStyle.Flat
            .FlatAppearance.BorderColor = Color.White
            .Text = Subtema & vbNewLine & "[ " & Tema & " ]"
            .Font = New Font("Arial", 11.2, FontStyle.Bold)
            .ForeColor = Color.Black
            .BackColor = Color.Transparent
            .TextAlign = ContentAlignment.MiddleLeft
            .UseVisualStyleBackColor = VisualStyles.BorderType.Rectangle
            .UseVisualStyleBackColor = VisualStyles.BackgroundColor.None
        End With

        AddHandler botoInfo.Click, AddressOf ClickEvent_botoVeureFitxa

        Dim caixaEtiquetes As New FlowLayoutPanel
        panell.Controls.Add(caixaEtiquetes)
        With caixaEtiquetes
            .Location = New System.Drawing.Point(botoInfo.Location.X +
botoInfo.Width + 10, botoInfo.Location.Y)
            .Width = panell.Width - 390
            .Height = 65
            .Anchor = AnchorStyles.Top
            .BorderStyle = BorderStyle.None

```

```

End With

Dim label As New Label
With label
    .Font = New Font("Arial", 10, FontStyle.Bold)
    .Text = "Etiquetes relacionades: "
    .AutoSize = True
    .Margin = New Padding(5, 3, 1, 3)
End With
caixaEtiquetes.Controls.Add(label)

etiquetes = Split(taula(i)(3), ", ", -1,
Microsoft.VisualBasic.CompareMethod.Binary)

Dim j As Integer = 0
While j < etiquetes.Count
    llistaEtiquetes.Add(etiquetes(j))

    j = j + 1
End While

Dim k As Integer = 0
While k < llistaEtiquetes.Count
    Dim etiqueta_k As New LinkLabel
    caixaEtiquetes.Controls.Add(etiqueta_k)
    With etiqueta_k
        .AutoSize = True
        .Margin = New Padding(5, 3, 5, 3)
        .Font = New Font("Arial", 10, FontStyle.Italic)
        .Text = llistaEtiquetes(k).ToString
        .Name = llistaEtiquetes(k).ToString
    End With

    Dim etiqueta_k_norm As String
    etiqueta_k_norm = NormalitzarString(etiqueta_k.Text)

    If cercaMod = "" Then

Else
    If nParaules > 1 Then
        j = 0
        While j < nParaules
            If LCase(etiqueta_k_norm).Contains(LCase(w_mod(j)))

Then
                etiqueta_k.LinkColor = Color.Green
            End If

            j = j + 1
        End While

    Else

        If LCase(etiqueta_k_norm).Contains(LCase(cercaMod)) Then
            etiqueta_k.LinkColor = Color.Green
        End If
    End If

End If

```

```

        AddHandler etiqueta_k.LinkClicked, AddressOf
ClickEvent_CercarEtiquetaClickada

        k = k + 1
    End While

    Dim botoPlus As New PictureBox
    panell.Controls.Add(botoPlus)
    With botoPlus
        .Name = i.ToString & "-botoPlus"
        .SetBounds(caixaEtiquetes.Location.X + caixaEtiquetes.Width +
10, caixaEtiquetes.Location.Y + caixaEtiquetes.Height - 25, 20, 20)
        .Image = imatge_plus
        .Anchor = AnchorStyles.Right
    End With

    AddHandler botoPlus.Click, AddressOf ClickEvent_ExpandirPanell

    Dim botoMinus As New PictureBox
    panell.Controls.Add(botoMinus)
    With botoMinus
        .Name = i.ToString & "-botoMinus"
        .SetBounds(caixaEtiquetes.Location.X + caixaEtiquetes.Width +
10, caixaEtiquetes.Location.Y + caixaEtiquetes.Height - 25, 20, 20)
        .Image = imatge_minus
        .Anchor = AnchorStyles.Right
        .SendToBack()
        .Visible = False
    End With

    AddHandler botoMinus.Click, AddressOf ClickEvent_ContraurePanell

    If caixaEtiquetes.Controls.Item(llistaEtiquetes.ElementAt(k -
1)).Location.Y < caixaEtiquetes.Height Then
        botoPlus.Visible = False
        botoMinus.Visible = False
    End If

    i = i + 1
    y = y + panell.Height + 10
    llistaPanells.Add(panell)
    llistaCaixesEtiquetes.Add(caixaEtiquetes)
    llistaBotonsPlus.Add(botoPlus)
    llistaBotonsMinus.Add(botoMinus)
    Array.Clear(etiquetes, 0, etiquetes.Length)
    llistaEtiquetes.Clear()
    Me.BringToFront()
    Panel1.Focus()
End While

AfegirSuggeriments(y, i)

If llistaCerques.Count > 1 Then
    botoEnrere.Visible = True
Else
    botoEnrere.Visible = False
End If

End If

```

```

End Sub

Private Sub ClickEvent_CercaAnterior(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    Dim j As Integer = llistaCerques.Count - 1
    llistaCerques.RemoveAt(j)
    Inicialitzar(llistaCerques.ElementAt(j - 1))
End Sub

''' Funció que crea un darrer panell amb els registres que no han sortit com
''' _a coincidents però que han estat consultats per altres usuaris navegant
''' _per etiquetes havent realitzat la mateixa cerca.
Private Sub AfegirSuggeriments(ByVal y As Integer, ByVal j As Integer)
    Dim dtview = New DataView(taulaRealim, " Cerca = '" & cercaNorm & "'",
"Valor DESC", DataViewRowState.CurrentRows)

    Dim taulaSuggeriments As New DataTable
    With taulaSuggeriments
        .Columns.Add("Cerca", GetType(String))
        .Columns.Add("Índex", GetType(Integer))
        .Columns.Add("Valor", GetType(Double))
    End With

    taulaSuggeriments = dtview.ToTable()

    Dim x As Integer = 0
    Dim i As Integer = 0

    Dim panell As New Panel
    Panel1.Controls.Add(panell)
    panell.Visible = False
    With panell
        .Name = j.ToString & "-Panell"
        .Location = New System.Drawing.Point(10, y)
        .Width = Panel1.Width - 50
        .AutoSizeMode = AutoSizeMode.GrowOnly
        .Height = 125
        .BorderStyle = BorderStyle.FixedSingle
        .Show()
    End With
    llistaPanells.Add(panell)

    Dim labelSuggeriments As New Label
    panell.Controls.Add(labelSuggeriments)
    With labelSuggeriments
        .Location = New System.Drawing.Point(12, 5)
        .Text = "Altres usuaris que han realitzat la mateixa cerca també han
consultat: "
        .Font = New Font("Arial", 10.5, FontStyle.Regular)
        .AutoSize = True
    End With

    While i < taulaSuggeriments.Rows.Count And x < 500
        Dim indexRegistre As Integer = taulaSuggeriments(i)(1)
        Dim registreEsResultat As Boolean = False

        For Each row As DataRow In taula.Rows
            If row(0) = indexRegistre Then

```

```

        registreEsResultat = True
        Exit For

    Else
        End If
    Next

    If registreEsResultat Then

    Else
        For Each rowInv As DataRow In taulaInventari.Rows
            If rowInv(0) = indexRegistre Then
                Dim Subtema As String = rowInv(2).ToString
                Dim Info As String = rowInv(4).ToString

                Dim registreNou As New RichTextBox
                panell.Controls.Add(registreNou)
                With registreNou
                    .Location = New System.Drawing.Point(x + 12,
labelSuggeriments.Height + 20)
                    .Width = 140
                    .Height = 50
                    .BorderStyle = BorderStyle.None
                    .Text = Subtema
                    .Select(0, Subtema.Length + 1)
                    .SelectionFont = New Font("Arial", 10, FontStyle.Bold)
                    .SelectionColor = Color.DarkRed
                    .SelectionAlignment = HorizontalAlignment.Center
                    .ReadOnly = True
                End With

                Dim botoVeureSuggerencia As New Button
                panell.Controls.Add(botoVeureSuggerencia)
                With botoVeureSuggerencia
                    .Name = Info
                    .Location = New System.Drawing.Point(x + 32,
registreNou.Location.Y + registreNou.Height + 3)
                    .Width = registreNou.Width - 40
                    .Height = 28
                    .Text = "Veure Fitxa"
                    .FlatStyle = FlatStyle.Flat
                    .BackColor = Color.WhiteSmoke
                End With

                AddHandler botoVeureSuggerencia.Click, AddressOf
ClickEvent_veureRegistreSuggerit

                panell.Visible = True
                x = x + registreNou.Width + 10
                Exit For
            End If
        Next
    End If

    i = i + 1
End While

If x = 0 Then panell.Visible = False

```

End Sub

```

'*****
'*****

'*****
'***** FUNCIONS QUE TRACTEN LA INFORMACIÓ QUE REALIMENTARÀ EL CERCADOR *****
'*****
'*** Funció que calcula, per la paraula cercada, el valor que es realimentarà
'***_dels registres consultats.
Public Sub CalculClics()
    If clics > 0 Then
        If clics = 1 Then
            valor = 1
        Else
            valor = 1 / clics
        End If

        Realimentar(valor)
        clics = 0
        llistaClics.Clear()
    End If
End Sub

'*** Funció que fa la realimentació de la BBDD.
Public Sub Realimentar(ByVal valor As Double)
    Dim existeix As Boolean = False
    Dim valorA As Double

    For Each item As Integer In llistaClics

        For Each row As DataRow In taulaRealim.Rows

            If row(0) = cercaNorm Then

                If row(1) = item Then
                    valorA = valor + row(2)
                    Actualitzar(item, valorA)
                    existeix = True
                    Exit For
                End If

            End If
        Next

        If Not existeix Then
            Insertar(item)
        End If

        Actualitzar_taulaRealim()
    Next
End Sub

'*** Funció que inicia o reinicia la taula de realimentació.
Public Sub Actualitzar_taulaRealim()
    Try
        taulaRealim.Columns.Add("Cerca", GetType(String))
        taulaRealim.Columns.Add("Index", GetType(Integer))
        taulaRealim.Columns.Add("Valor", GetType(Double))
    
```

```

Catch ex As Exception

End Try
taulaRealim.Reset()

'* Emplenar una Datatable amb la info d'antigues cerques per
'*_autorealimentar-se i millorar l'experiència.
da4 = New OleDbDataAdapter(consultaSQL_taulaRealim, connection4)
da4.Fill(taulaRealim)
End Sub

'*** Esdeveniment quan es tanca l'aplicació.
Private Sub Cercador_FormClosing(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
    If clics > 0 Then
        CalculClics()
    End If
End Sub

'*****
'*****
'*****
'***** FUNCIONS QUE ACTUALITZEN LA INFORMACIÓ DE LES BBDD *****
'*****

Public Sub Actualitzar(ByVal indexB As Integer, ByVal valorA As Double)
    Dim iSql As String
    If ORAtrue_ACCfalse = True Then
        Dim valorAstr As String = CStr(valorA)

        Dim numE, numD As Integer
        numE = InStr(valorAstr, ",") - 1
        numD = InStrRev(valorAstr, ",") - 1

        Dim val As Object

        If numE = -1 Then
            val = valorA
        Else
            val = Microsoft.VisualBasic.Left(valorAstr, numE) & "." &
Microsoft.VisualBasic.Right(valorAstr, numD)

        End If

        iSql = "UPDATE REALIMENTACIO SET valor =" & val & " WHERE cerca =" &
cercaOriginal & "' AND indexB =" & indexB
        Dim cmd As New OleDb.OleDbCommand(iSql, connDB)

        connDB.Open()
        cmd.ExecuteNonQuery()
        connDB.Close()

    Else : ORAtrue_ACCfalse = False
        MsgBox(valorA & vbNewLine & cercaOriginal & vbNewLine & indexB)
        iSql = "UPDATE REALIMENTACIO SET VALOR = @VALOR WHERE CERCA = @CERCA AND
INDEXB = @INDEXB"
        Dim cmd As New OleDb.OleDbCommand(iSql, connDB)

```

```

        '* Utilitzar paràmetres
        cmd.Parameters.AddWithValue("@VALOR", valorA)
        cmd.Parameters.AddWithValue("@CERCA", cercaOriginal)
        cmd.Parameters.AddWithValue("@INDEXB", indexB)

        connDB.Open()
        cmd.ExecuteNonQuery()
        connDB.Close()

    End If
End Sub

Public Sub Insertar(ByVal indexB As Integer)
    Dim iSql As String

    If ORATrue_ACCfalse = True Then
        Dim valorAstr As String = CStr(valor)

        Dim numE, numD As Integer
        numE = InStr(valorAstr, ",") - 1
        numD = InStrRev(valorAstr, ",") - 1

        Dim val As Object
        If numE = -1 Then
            val = valor
        Else
            val = Microsoft.VisualBasic.Left(valorAstr, numE) & "." &
Microsoft.VisualBasic.Right(valorAstr, numD)

        End If
        iSql = "INSERT INTO REALIMENTACIO (Cerca, IndexB, Valor) VALUES('" &
cercaOriginal & "', " & indexB & ", " & val & ")"
        Dim cmd As New OleDb.OleDbCommand(iSql, connDB)

        connDB.Open()
        cmd.ExecuteNonQuery()
        connDB.Close()

    Else : ORATrue_ACCfalse = False
        iSql = "INSERT INTO REALIMENTACIO (Cerca, IndexB, Valor) VALUES(" &
cercaOriginal & ", " & indexB & ", " & valor & ")"
        Dim cmd As New OleDb.OleDbCommand(iSql, connDB)

        '* Utilitzar paràmetres
        cmd.Parameters.AddWithValue("@cerca", cercaOriginal)
        cmd.Parameters.AddWithValue("@indexB", indexB)
        cmd.Parameters.AddWithValue("@valor", valor)

        connDB.Open()
        cmd.ExecuteNonQuery()
        connDB.Close()

    End If
End Sub

```

```

'*****
'*****

```



```

'*****
'***** FUNCIONS QUE RETORNEN L'String ENTRANT AMB LES MODIFICACIONS PERTINENTS **
'*****

'*** Funció que retorna un string sense accents ni dièresis.
Function NormalitzarString(ByVal stringPerNorm)
    Dim stringNorm As String = stringPerNorm
    Dim llista_strings() As String

    If (String.IsNullOrEmpty(stringPerNorm)) Then
        stringNorm = ""
    Else
        llista_strings = Split(stringPerNorm, " ")

        Dim i, j As Integer
        Dim charactersNoPermesos(), charactersPermesos() As String
        charactersNoPermesos = {"á", "é", "í", "ó", "ú", "à", "è", "ì", "ò", "ù",
"a", "ë", "ï", "ö", "ü"}
        charactersPermesos = {"a", "e", "i", "o", "u", "a", "e", "i", "o", "u",
"a", "e", "i", "o", "u"}

        j = 0
        While j < llista_strings.Length
            For Each lletra As Char In llista_strings(j)

                i = 0
                While i < charactersNoPermesos.Length
                    If lletra = charactersNoPermesos(i) Then
                        stringNorm = Replace(llista_strings(j),
charactersNoPermesos(i), charactersPermesos(i))
                        llista_strings(j) = stringNorm

                    End If

                    i = i + 1
                End While

            Next

            j = j + 1
        End While

        stringNorm = Join(llista_strings, " ")

    End If

    Return LCase(stringNorm)
End Function

'*** Funció que retorna un string amb el problema de l'apostrofe en una
'***_consulta SQL solucionat.
Function TreureApostrof(ByVal paraula_in As String)
    Dim paraula_out As String
    paraula_out = Replace(paraula_in, "'", "'")
    Return paraula_out
End Function

```

```

*** Funció que retorna un string sense preposicions intermèdies.
Function TreurePreposicions(ByVal string_in)
    Dim string_out As String = string_in

    If (String.IsNullOrEmpty(string_in)) Then
        string_out = ""
    Else
        Dim i, j As Integer
        Dim preposicions() As String
        preposicions = {" en el ", " de la ", " el ", " els ", " la ", " les ",
" de ", " del ", " dels ", " a ", " en ", " per ", " i ", " o "}

        i = 0
        While i < preposicions.Length
            If string_in.Contains(preposicions(i)) Then
                j = string_in.IndexOf(preposicions(i))
                string_out = string_in.Remove(j + 1, Len(preposicions(i)) - 1)
            End If

            i = i + 1
        End While

    End If

    Return string_out
End Function

*** Funció que retorna la cerca sense més d'un espai entre paraules.
Function TreureEspaisEnBlanc(ByVal string_in As String)
    Dim string_out As String = string_in
    Dim i, j, k As Integer

    If (String.IsNullOrEmpty(string_in)) Then
        string_out = ""
    Else
        i = 0
        While i < string_out.Length

            If string_out.Chars(i) = " " Then
                k = 1
                j = i + 1
                While j < string_out.Length

                    If string_out.Chars(j) = " " Then
                        k = k + 1
                    Else

                        If k > 1 Then
                            string_out = string_out.Remove(i + 1, k - 1)

                        End If
                        Exit While

                    End If

                    j = j + 1
                End While
            End If

            i = i + 1
        End While
    End If

```

```

        End While
    End If

    i = i + 1
End While

End If

Return string_out
End Function

'*****
'*****
'*****
'***** ALGORISMES PER PRIMITIVITZAR O BUSCAR EQUIVALENTS D'UNA PARAULA *****
'*****
'*** Algorisme per trobar l'arrel o primitiva de la paraula introduïda.
Function FerArrel(ByVal b As String)
    Dim R1 As String = 0
    Dim R2 As String = 0
    Dim RV As String = 0
    Dim Arrel As String = b
    Dim l As Integer = Len(b)

    If l = 3 Or l = 2 Or l = 1 Then
        '* L'arrel és la pròpia paraula.
        Arrel = b
    End If

    If l = 4 Then
        '* L'arrel és la pròpia paraula a menys que acabi en "s".
        Dim TestPos As Integer
        TestPos = InStr(4, b, "s", CompareMethod.Text)
        If TestPos = 4 Then
            Arrel = Microsoft.VisualBasic.Left(b, 3)
        Else
            Arrel = b
        End If
    End If

    If l >= 5 Then
        FerR2(b, l, R1, R2)

        '* Si R2 és 0, es farà RV per trobar l'arrel en aquests casos.
        If R2 = "0" Then
            FerRV(b, R1, RV)
            Arrel = b.Remove(1 - Len(RV))
        Else '* Si R2 no és 0, ja s'ha trobat l'arrel.
            Arrel = b.Remove(1 - Len(R2))
        End If
    End If

    Return Arrel
End Function

```

```

    *** Subfunció de FerArrel que extreu un substring de la paraula entrant.
    Private Sub FerR2(ByVal w As String, ByVal l As Integer, ByRef R1 As String,
ByRef R2 As String)
        Dim i, voc, cons As Integer
        Dim TestPos_v, TestPos_c As Integer
        Dim Trobat As Boolean
        Dim vocal(), consonant() As String

        vocal = {"a", "à", "e", "è", "é", "i", "í", "o", "ò", "ó", "u", "ú", "ü"}
        consonant = {"b", "c", "d", "f", "g", "h", "j", "k", "l", "m", "n", "ñ",
"p", "q", "r", "s", "t", "v", "w", "x", "y", "z"}

        i = 1
        Trobat = False

        *****
        '* Es buscarà primer un substring per extreure de la paraula inicial (R1)
        '* _i posteriorment (si és possible) un altre (R2).
        '* Després, aquests substrings se sumaran i es restaran de la paraula
        '* _inicial.
        '* Primer es busca R1 ja que R2 es trobarà sobre R1.
        *****
        While i + 1 < l And Not Trobat
            voc = 0
            cons = 0
            TestPos_v = 0
            TestPos_c = 0
            While cons < consonant.Length And voc < vocal.Length And Not Trobat

                '* Cerca de la vocal en la posició i.
                If TestPos_v = 0 Then
                    TestPos_v = InStr(i, w.Remove(i + 1), vocal(voc),
CompareMethod.Text)
                End If
                '* Cerca de la consonant en la posició i+1.
                If TestPos_c = 0 Then
                    TestPos_c = InStr(i + 1, w.Remove(i + 1), consonant(cons),
CompareMethod.Text)
                End If

                '* Si es tenen dues posicions consecutives de vocal-consonant ja
                '* _es pot sortir del While.
                If (TestPos_v + 1 = TestPos_c) Then
                    R1 = Microsoft.VisualBasic.Right(w, l - (i + 1))
                    Trobat = True
                Else
                    End If

                '* Condicions per sortir del While.
                If TestPos_v = 0 Then
                    voc = voc + 1
                End If

                If TestPos_c = 0 Then
                    cons = cons + 1
                End If

            End While

        End While

```

```

        i = i + 1
    End While

    '* Ara, es busca R2 sobre R1.

    i = 1
    Trobat = False

    '* Si R1 no consta de més de 2 caràcters, R2 serà la paraula b.
    If Len(R1) > 2 Then
        '*Es torna a fer el While anterior amb la diferència de canviar 'b'
        '*_per 'R1'.
        While i < Len(R1) - 1 And Not Trobat
            voc = 0
            cons = 0
            TestPos_v = 0
            TestPos_c = 0
            While cons < consonant.Length And voc < vocal.Length And Not Trobat

                '* Cerca de la vocal en la posició i.
                If TestPos_v = 0 Then
                    TestPos_v = InStr(i, R1.Remove(i + 1), vocal(voc),
CompareMethod.Text)
                End If
                '* Cerca de la consonant en la posició i+1.
                If TestPos_c = 0 Then
                    TestPos_c = InStr(i + 1, R1.Remove(i + 1), consonant(cons),
CompareMethod.Text)
                End If

                '* Si es tenen dues posicions consecutives de vocal-consonant
                '*_ja es pot sortir del While.
                If (TestPos_v + 1 = TestPos_c) Then
                    R2 = Microsoft.VisualBasic.Right(R1, Len(R1) - (i + 1))
                    Trobat = True
                Else
                    End If

                '* Condicions per sortir del While.
                If TestPos_v = 0 Then
                    voc = voc + 1
                End If

                If TestPos_c = 0 Then
                    cons = cons + 1
                End If

            End While

            i = i + 1
        End While
    Else
        R2 = 0
    End If
End Sub

Private Sub FerRV(ByVal b As String, ByVal R1 As String, ByRef RV As String)
    Dim a As String

```

```

a = b.Remove(b.Length - Len(R1))

If Len(a) <= 2 Then
    RV = a

ElseIf Len(a) > 2 Then
    '* Exemple creueta per inspirar-me!
    RV = Microsoft.VisualBasic.Right(b, b.Length - 3)

End If
End Sub

'*** Funció que retorna (si existeix) una paraula equivalent de la que entra.
Function Tesauro(BYVal paraula As String)
    Dim equiv As String = ""
    Dim i As Integer = 0

    Dim taulaEquivalencies As New DataTable
    '* Emplenar un Datatable amb la info d'equivalències per fer el Tesauro.
    da5 = New OleDbDataAdapter(consultaSQL_tesauro, connection5)
    da5.Fill(taulaEquivalencies)

    While i < nParaules
        For Each row As DataRow In taulaEquivalencies.Rows

            If LCase(row(2)).Contains(LCase(paraula)) Then
                equiv = row(1)

            Else
            End If

        Next

        i = i + 1
    End While

    If equiv = "" Then
        Return paraula

    Else
        Return equiv

    End If
End Function

'*****
'*****

'*****
'***** ESDEVENIMENTS QUE INICIEN UNA CERCA, MOSTREN UNA FITXA O OBREN L'ARBRE ***
'*****

'*** Esdeveniment quan es clica una etiqueta linkada de la caixaEtiquetes.
Public Sub ClickEvent_CercarEtiquetaClickada(BYVal sender As System.Object,
BYVal e As System.EventArgs)
    Dim linkEtiqueta As String
    linkEtiqueta = sender.Text
    Inicialitzar(linkEtiqueta)
End Sub

```

```

    *** Esdeveniment quan es clica una de les possibles referències de la
    ***_informació disposades en la seva classificació en l'Arbre Conductor.
Public Sub ClickEvent_CercarReferenciaClickada(ByVal sender As System.Object,
ByVal e As System.EventArgs)
    cercaOriginal = LCase(NormalitzarString(sender.Text))
    cercaNorm = LCase(NormalitzarString(sender.Text))
    ComboBox1.Text = ""

    Try
        taula.Columns.Add("Índex", GetType(Integer))
        taula.Columns.Add("Tema", GetType(String))
        taula.Columns.Add("Subtema", GetType(String))
        taula.Columns.Add("Etiquetes", GetType(String))
        taula.Columns.Add("Informació", GetType(String))
        taula.Columns.Add("Valor", GetType(Double))

    Catch ex As Exception
        If taula.Rows.Count > 0 Then
            Panel1.Controls.Clear()
            taula.Rows.Clear()
            llistaPanells.Clear()
            llistaCaixesEtiquetes.Clear()
            llistaBotonsPlus.Clear()
            llistaBotonsMinus.Clear()
        End If

    End Try

    For Each row As DataRow In taulaInventari.Rows
        If LCase(row(4)) = LCase(sender.Text) Then
            taula.Rows.Add(New Object() {row(0), row(1), row(2), row(3), row(4),
1})

        End If
    Next

    AfegirRegistres()
End Sub

    *** Esdeveniment que inicia l'acció per veure la fitxa informativa del
    ***_registre clicat.
Private Sub ClickEvent_botoVeureFitxa(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    Dim nDigits As String
    Dim i As Integer

    nDigits = InStr(sender.Name, "-",
Microsoft.VisualBasic.CompareMethod.Binary) - 1
    i = CInt(Microsoft.VisualBasic.Left(sender.Name, nDigits))

    llistaClics.Add(taula(i)(0))
    clics = clics + 1

    Informacio.MostrarFitxaInformacio(taula(i)(4))
End Sub

    *** Esdeveniment que inicia l'acció per veure la fitxa informativa del
    ***_registre suggerit clicat.

```

```

Private Sub ClickEvent_veureRegistreSuggerit(ByVal sender As System.Object,
ByVal e As System.EventArgs)
    Informacio.MostrarFitxaInformacio(sender.Name.ToString)
End Sub

'*** Esdeveniment que inicia l'acció per visualitzar l'Arbre Conductor.
Private Sub BotoArbre_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BotoArbre.Click
    Actualitzar_taulaRealim()
    With ArbreConductor
        .Show()
        .Focus()
        .Label1.Visible = True
        .Label1.SendToBack()
        .Label2.Visible = False
    End With
End Sub

'*****
'*****

'*****
'***** COMPORTAMENT DE LA INTERFÍCIE EN CLICAR ELS BOTONS EXPANDIR/CONTRAURE *****
'*****

'*** Acció que expandeix la caixaEtiquetes quan es clica el botoPlus.
Private Sub ClickEvent_ExpandirPanell(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    sender.Visible = False

    Dim i As Integer
    Dim nDigits As Integer

    nDigits = InStr(sender.Name, "-",
Microsoft.VisualBasic.CompareMethod.Binary) - 1
    i = CInt(Microsoft.VisualBasic.Left(sender.Name, nDigits))

    With llistaCaixesEtiquetes.ElementAt(i)
        .Height = .Controls.Item(.Controls.Count - 1).Location.Y + 20
    End With

    With llistaPanells.ElementAt(i)
        .Height = llistaCaixesEtiquetes.ElementAt(i).Height + 15
    End With

    With llistaBotonsMinus.ElementAt(i)
        .Location = New
System.Drawing.Point(llistaBotonsPlus.ElementAt(i).Location.X,
llistaCaixesEtiquetes.ElementAt(i).Location.Y +
llistaCaixesEtiquetes.ElementAt(i).Height - 25)
        .BringToFront()
        .Visible = True
    End With

    RecolocarPanells_Avall(i)
End Sub

'*** Acció que contrau la caixaEtiquetes quan es clica el botoMinus.
Private Sub ClickEvent_ContraurePanell(ByVal sender As System.Object, ByVal e As
System.EventArgs)

```



```

    Dim i As Integer
    Dim nDigits As Integer

    nDigits = InStr(sender.Name, "-",
Microsoft.VisualBasic.CompareMethod.Binary) - 1
    i = CInt(Microsoft.VisualBasic.Left(sender.Name, nDigits))

    llistaCaixesEtiquetes.ElementAt(i).AutoScroll = False
    Panel1.Focus()
    llistaBotonsPlus.ElementAt(i).Visible = True
    With llistaBotonsMinus.ElementAt(i)
        .Location = New
System.Drawing.Point(llistaBotonsPlus.ElementAt(i).Location.X,
llistaBotonsPlus.ElementAt(i).Location.Y)
        .SendToBack()
        .Visible = False
    End With

    llistaCaixesEtiquetes.ElementAt(i).Height = 65
    llistaPanells.ElementAt(i).Height =
llistaCaixesEtiquetes.ElementAt(i).Height + 15

    RecolocarPanells_Amunt(i)
End Sub

'*** Funció que recoloca els panells que estan per sota del panell "i".
Private Sub RecolocarPanells_Avall(ByVal i As Integer)
    Dim y As Integer
    y = llistaPanells.ElementAt(i).Location.Y +
llistaPanells.ElementAt(i).Height + 10

    While i < llistaPanells.Count - 1
        llistaPanells.ElementAt(i + 1).Location = New
System.Drawing.Point(llistaPanells.ElementAt(i).Location.X, y)
        y = y + llistaPanells.ElementAt(i + 1).Height + 10
        i = i + 1
    End While
End Sub

'*** Funció que recoloca els panells que estan per sobre del panell "i".
Private Sub RecolocarPanells_Amunt(ByVal i As Integer)
    Dim y As Integer
    y = llistaPanells.ElementAt(i).Location.Y +
llistaPanells.ElementAt(i).Height + 10

    While i < llistaPanells.Count - 1
        llistaPanells.ElementAt(i + 1).Location = New
System.Drawing.Point(llistaPanells.ElementAt(i).Location.X, y)
        y = y + llistaPanells.ElementAt(i + 1).Height + 10
        i = i + 1
    End While
End Sub

'*****
'*****

'*****
'***** COMPORTAMENT AL MAXIMITZAR/MINIMITZAR LA FINESTRA *****
'*****

```

```

    *** Tot el que s'ha de tenir en compte quan es maximitza/minimitza la
    ***_finestra.
    Private Sub Cercador_Resize(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Resize

        Panel1.Visible = False

        For Each panell In llistaPanells
            With panell
                .Width = Panel1.Width - 50
            End With
        Next

        Dim i As Integer = 0
        For Each caixaEtiquetes In llistaCaixesEtiquetes

            With caixaEtiquetes
                .Width = Panel1.Width - 440
                .Location = New System.Drawing.Point(330, caixaEtiquetes.Location.Y)

                If .Height = 65 Then
                    llistaBotonsMinus.ElementAt(i).Visible = False
                    If .Height < .Controls.Item(.Controls.Count - 1).Location.Y Then
                        llistaBotonsPlus.ElementAt(i).Visible = True

                    Else
                        llistaBotonsPlus.ElementAt(i).Visible = False

                    End If

                Else
                    llistaBotonsMinus.ElementAt(i).Visible = True
                    .Height = .Controls.Item(.Controls.Count - 1).Location.Y + 18

                    If .Height < 65 Then
                        .Height = 65
                        llistaBotonsMinus.ElementAt(i).Visible = False
                        llistaBotonsPlus.ElementAt(i).Visible = False

                    ElseIf .Height = 65 Then
                        llistaBotonsMinus.ElementAt(i).Visible = False
                        llistaBotonsPlus.ElementAt(i).Visible = False

                    End If

                    llistaPanells.ElementAt(i).Height = .Height + 15
                    llistaBotonsMinus.ElementAt(i).Location = New
System.Drawing.Point(llistaBotonsMinus.ElementAt(i).Location.X, .Location.Y +
.Height - 25)
                    RecolocarPanells_Amunt(i)
                    RecolocarPanells_Avall(i)

                End If

            End With

            i = i + 1
        Next

```

```

        ComboBox1.SelectionStart = ComboBox1.Text.Length

        With Panel1
            .Visible = True
            .Focus()
            .AutoScrollPosition = New
System.Drawing.Point(Panel1.AutoScrollPosition.X, pos_scroll)
        End With
    End Sub

    '*** Permet mantenir la mateixa alçada del scroll després de
    '*** _maximitzar/minimitzar.
    Private Sub Panel1_MouseWheel(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Panel1.MouseWheel
        pos_scroll = -1 * Panel1.AutoScrollPosition.Y
    End Sub

    '*** Permet mantenir la mateixa alçada del scroll després de
    '*** _maximitzar/minimitzar.
    Private Sub Panel1_Scroll(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.ScrollEventArgs) Handles Panel1.Scroll
        pos_scroll = -1 * Panel1.AutoScrollPosition.Y
        Panel1.Focus()
    End Sub

    '*****
    '*****
End Class

```

## Part 2: Fitxa informativa

```

Public Class Informacio
    Private propietats_detall As New List(Of String)

    Private Sub Informació_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Me.Icon = Cercador.Icon
    End Sub

    '*** Finestra amb la informació detallada del registre.
    Public Sub MostrarFitxaInformacio(ByVal ubiImportada As String)

        For Each row As DataRow In Cercador.taulaInformacio.Rows
            If row(1) = ubiImportada Then
                Dim j As Integer = 0

                While j < Cercador.propietats_info.Count

                    If row(j + 2).ToString = "" Then
                        propietats_detall.Add("")
                    Else
                        propietats_detall.Add(row(j + 2))
                    End If
                End While
            End If
        End For
    End Sub
End Class

```

```

        j = j + 1
    End While
Exit For

End If
Next

With RichTextBox1
    .ReadOnly = True
    .Text = Cercador.Titol_fitxaInfo & ": " & ubiImportada & vbNewLine
    .Font = New Font("Arial", 11, FontStyle.Bold)

    If propietats_detall.Count > 0 Then
        Dim i As Integer = 0
        While i < Cercador.propietats_info.Count
            If propietats_detall.ElementAt(i).Length > 1 Then
                Cercador.propietats_info.ElementAt(i).ToString & ": " &
                propietats_detall.ElementAt(i).ToString)
                .Select(.TextLength -
                Cercador.propietats_info.ElementAt(i).Length - propietats_detall.ElementAt(i).Length
                - 2, Cercador.propietats_info.ElementAt(i).Length + 1)
                .SelectionColor = Color.DarkRed
            End If

            i = i + 1
        End While

        .Select(0, Cercador.Titol_fitxaInfo.Length + ubiImportada.Length +
2)
        .SelectionFont = New Font("Arial", 13, FontStyle.Bold)
        .SelectionStart = .TextLength

        If propietats_detall.ElementAt(0).Length > 1 Then
            Me.Show()
            Me.BringToFront()
        Else
            MsgBox("Actualment no existeix informació sobre " &
ubiImportada)
        End If

        Else
            MsgBox("Encara no hi ha informació disponible per a aquest
registre")
        End If

    End With
End Sub

'*** La finestra amb la informació disponible del registre clicat es tanca.
Private Sub BotoFitxa_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BotoFitxa.Click
    RichTextBox1.Clear()
    propietats_detall.Clear()
    Me.Close()
End Sub

End Class

```

### Part 3: Arbre conductor

```
Imports System.Data.OleDb

Public Class ArbreConductor
    Private da As OleDbDataAdapter

    Public llistaPanells As New List(Of Panel)

    '* Botó per tornar a l'arbre general.
    Dim botoEnrere As New PictureBox
    Dim FLpanel As New FlowLayoutPanel

    '* Quan es carrega la finestra de l'Arbre Conductor.
    Public Sub ArbreConductor_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Me.Icon = Cercador.Icon
        PanellArbre.Visible = True
        PanellArbre.BringToFront()
    End Sub

    '* ** Classificació segons el tipus de servei públic (Classificació OpenData
    '* ** _Barcelona).
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        OrganitzarFinestra()

        Dim taulaOpenData As New DataTable
        '* Emplenar el Datatable amb la info de la Classificació OpenData.
        da = New OleDbDataAdapter(Cercador.consultaSQL_OpenData,
Cercador.connection6)
        da.Fill(taulaOpenData)

        Dim label As New Label
        Panel1.Controls.Add(label)
        With label
            .Text = "Aquestes són les principals etiquetes classificades en 5 grans
temes: "
            .Font = New Font("Arial", 11.5, FontStyle.Bold)
            .Location = New System.Drawing.Point(20, 15)
            .AutoSize = True
        End With

        Panel1.Controls.Add(FLpanel)
        With FLpanel
            .Dock = DockStyle.Bottom
            .Height = Panel1.Height - 40
        End With

        Dim i As Integer = 0
        Dim j As Integer = 0

        While j < taulaOpenData.Columns.Count

            '* panell Tema OpenData.
            Dim panell As New FlowLayoutPanel
            FLpanel.Controls.Add(panell)
            With panell
```

```

        .Width = 350
        .Height = 150
        .Margin = New Padding(20, 20, 10, 10)
    End With

    '* label del panell Tema OpenData.
    Dim labelpanell As New Label
    panell.Controls.Add(labelpanell)
    With labelpanell
        .Text = taulaOpenData.Columns(j).ColumnName
        .Font = New Font("Arial", 12, FontStyle.Bold)
        .ForeColor = Color.DarkRed
        .Width = 350
        .Height = 16
    End With

    '* Etiquetes del Tema OpenData.
    i = 0
    While i < taulaOpenData.Rows.Count
        If taulaOpenData(i)(j).ToString.Length > 1 Then
            Dim botoetiqueta As New Button
            panell.Controls.Add(botoetiqueta)
            With botoetiqueta
                .Text = taulaOpenData(i)(j).ToString
                .Font = New Font("Arial", 11, FontStyle.Bold)
                .AutoSize = True
                .Margin = New Padding(8, 6, 8, 4)
                .FlatStyle = FlatStyle.Flat
                .BackColor = Color.WhiteSmoke
            End With
            AddHandler botoetiqueta.Click, AddressOf
Cercador.ClickEvent_CercarEtiquetaClickada
        Else
            End If

            i = i + 1
        End While

        j = j + 1
    End While

    Panel1.Focus()
    CrearBotoEnrere()
End Sub

''' Classificació de totes les referències de la informació que hi ha a la base
'''_de dades.
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    OrganitzarFinestra()

    Dim llistaReferencies As New List(Of String)
    Dim i As Integer = 0

    For Each row As DataRow In Cercador.taulaInformacio.Rows
        llistaReferencies.Add(row(1))
    Next

```

```

Dim label As New Label
Panel1.Controls.Add(label)
With label
    .Text = "Aquestes són totes les referències: "
    .Font = New Font("Arial", 11.5, FontStyle.Bold)
    .Location = New System.Drawing.Point(20, 15)
    .AutoSize = True
End With

Dim panellBotonsReferencia As New FlowLayoutPanel
Panel1.Controls.Add(panellBotonsReferencia)
llistaPanells.Add(panellBotonsReferencia)
With panellBotonsReferencia
    .Location = New System.Drawing.Point(30, 60)
    .Width = Panel1.Width - 60
    .Height = Panel1.Height - 60
    .Anchor = AnchorStyles.Top
    .AutoScroll = True
End With

i = 0
While i < llistaReferencies.Count

    Dim botoRef As New Button
    panellBotonsReferencia.Controls.Add(botoRef)
    With botoRef
        .Text = llistaReferencies(i)
        .Font = New Font("Arial", 10.5, FontStyle.Bold)
        .FlatStyle = FlatStyle.Flat
        .BackColor = Color.WhiteSmoke
        .TextAlign = ContentAlignment.MiddleCenter
        .Width = 195
        .Height = 60
        .Margin = New Padding(10, 8, 10, 8)
    End With
    AddHandler botoRef.Click, AddressOf
Cercador.ClickEvent_CercarReferenciaClickada

    i = i + 1
End While

CrearBotoEnrere()
panellBotonsReferencia.Focus()
End Sub

'*** Classificació de tots els temes existents a la base de dades.
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    OrganitzarFinestra()

    Dim llistaTemes As New List(Of String)
    Dim i As Integer = 0

    '* Afegir temes a la llista.
    While i < Cercador.taulaInventari.Rows.Count
        Dim j As Integer = 0
        '* Si el Tema j no està a la llista, s'haurà recorregut tot el WHILE i
        '*_aleshores j = llistaTemes.Count.
        While j < llistaTemes.Count

```

```

        If Cercador.taulaInventari(i)(1) = llistaTemes(j) Then
            Exit While
        Else
            j = j + 1
        End If
    End While

    If j = llistaTemes.Count Then
llistaTemes.Add(Cercador.taulaInventari(i)(1))

        i = i + 1
    End While

    Dim label As New Label
    Panel1.Controls.Add(label)
    With label
        .Text = "Aquests són tots els temes: "
        .Font = New Font("Arial", 11.5, FontStyle.Bold)
        .Location = New System.Drawing.Point(20, 15)
        .AutoSize = True
    End With

    Dim panellBotonsTema As New FlowLayoutPanel
    Panel1.Controls.Add(panellBotonsTema)
    llistaPanells.Add(panellBotonsTema)
    With panellBotonsTema
        .Location = New System.Drawing.Point(30, 60)
        .Width = Panel1.Width - 60
        .Height = Panel1.Height - 60
        .Anchor = AnchorStyles.Top
        .AutoScroll = True
    End With

    i = 0
    While i < llistaTemes.Count

        Dim botoTema As New Button
        panellBotonsTema.Controls.Add(botoTema)
        With botoTema
            .Text = llistaTemes(i)
            .Font = New Font("Arial", 10.5, FontStyle.Bold)
            .FlatStyle = FlatStyle.Flat
            .BackColor = Color.WhiteSmoke
            .TextAlign = ContentAlignment.MiddleCenter
            .Width = 195
            .Height = 60
            .Margin = New Padding(10, 8, 10, 8)
        End With
        AddHandler botoTema.Click, AddressOf
Cercador.ClickEvent_CercarEtiquetaClickada

        i = i + 1
    End While

    CrearBotoEnrere()
    panellBotonsTema.Focus()
End Sub

'*** Classificació dels registres més consultats pels usuaris (més populars).

```



```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    OrganitzarFinestra()

    Dim dw As New DataView(Cercador.taulaRealim)
    dw.Sort = "Valor DESC"

    Dim llistaPopus As New List(Of Integer)
    Dim i As Integer = 0

    While llistaPopus.Count < 12
        Dim j As Integer = 0

        While j < llistaPopus.Count
            If dw(i)(1) = llistaPopus(j) Then
                Exit While
            Else
                j = j + 1
            End If
        End While

        If j = llistaPopus.Count Then llistaPopus.Add(dw(i)(1))

        i = i + 1
    End While

    Dim label As New Label
    Panel1.Controls.Add(label)
    With label
        .Text = "Aquestes són les etiquetes dels registres més buscats: "
        .Font = New Font("Arial", 11.5, FontStyle.Bold)
        .Location = New System.Drawing.Point(20, 15)
        .AutoSize = True
    End With

    Dim panellEtiquetesPopus As New FlowLayoutPanel
    Panel1.Controls.Add(panellEtiquetesPopus)
    llistaPanells.Add(panellEtiquetesPopus)
    With panellEtiquetesPopus
        .Location = New System.Drawing.Point(30, 60)
        .Width = Panel1.Width - 60
        .Height = Panel1.Height - 60
        .Anchor = AnchorStyles.Top
        .AutoScroll = True
    End With

    i = 0
    While i < llistaPopus.Count

        Dim text_etiqueta As String = "0"
        For Each row As DataRow In Cercador.taulaInventari.Rows
            If row(0) = llistaPopus(i) Then
                text_etiqueta = row(2).ToString
            End If
        Next

        If text_etiqueta = "0" Then
```

```

Else
    Dim botoPopu As New Button
    panellEtiquetesPopus.Controls.Add(botoPopu)
    With botoPopu
        .Text = text_etiqueta
        .Font = New Font("Arial", 10.5, FontStyle.Bold)
        .FlatStyle = FlatStyle.Flat
        .BackColor = Color.WhiteSmoke
        .TextAlign = ContentAlignment.MiddleCenter
        .Width = 195
        .Height = 60
        .Margin = New Padding(10, 8, 10, 8)
    End With
    AddHandler botoPopu.Click, AddressOf
Cercador.ClickEvent_CercarEtiquetaClickada

End If

i = i + 1
End While

CrearBotoEnrere()
panellEtiquetesPopus.Focus()
End Sub

'*** Esdeveniment al clicar el botó enrere.
Private Sub ClickEvent_TornarInici(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    Panel1.Controls.Clear()
    FLpanel.Controls.Clear()
    Panel1.SendToBack()
    Panel1.Visible = False
    PanellArbre.BringToFront()
    PanellArbre.Visible = True
End Sub

'*** Funció que fa totes les accions necessàries per mostrar en pantalla el que
cal.
Private Sub OrganitzarFinestra()
    PanellArbre.SendToBack()
    PanellArbre.Visible = False
    Panel1.BringToFront()
    Panel1.Visible = True
    Label1.SendToBack()
    Label1.Visible = False
    Label2.SendToBack()
    Label2.Visible = False
End Sub

'*** Es crea el botó enrere.
Private Sub CrearBotoEnrere()
    '* Botó per tornar a l'arbre general
    Panel1.Controls.Add(botoEnrere)
    With botoEnrere
        .Image = Cercador.imatge_back
        .Location = New System.Drawing.Point(Panel1.Width - 52, 8)
        .AutoSize = True
    End With
    AddHandler botoEnrere.Click, AddressOf ClickEvent_TornarInici

```

```
End Sub

'*** Tot el que s'ha de tenir en compte quan es maximitza/minimitza la finestra.
Private Sub ArboreConductor_Resize(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Resize

    botoEnrere.Location = New System.Drawing.Point(Panel1.Width - 52, 8)

    FLpanel.Height = Panel1.Height - 40

    For Each panell As Panel In llistaPanells
        With panell
            .Location = New System.Drawing.Point(30, 60)
            .Width = Panel1.Width - 60
            .Height = Panel1.Height - 60
        End With
    Next

End Sub

End Class
```

## Annex II: Fitxer “Cercador.ini”

A continuació es mostrarà el contingut del fitxer *.ini* que conté tota la parametrització externalitzada.

```
[ConnectionString]
Provider=Microsoft.ACE.OLEDB.15.0;          Data          Source=C:\Users\Josep
Solé\Desktop\bin_Cercador\BBDDcercador.accdb;Persist Security Info=False
Provider=MSDAORA.1;Password=VISTAMAP_P;User ID=VISTAMAP_U;Data Source=dsv1;Persist
Security Info=True
Data Source=dsv1;Initial Catalog=MSDAORA.1;User ID=VISTAMAP_U;Password=VISTAMAP_P;
```

```
[ImportTable]
title:taulaAutocompletar
con:1
```

```
select
    Etiquetes
from
    llistatEtiquetes
;
```

```
[ImportTable]
title:taulaInventari
con:1
```

```
select
    *
from
    nouInventari
order by
    IdPropi asc
;
```

```
[ImportTable]
title:taulaInformacio
con:1
```

```
select
    *
from
    informacio
order by
    Resultat desc
;
```

```
[ImportTable]
title:taulaRealimentacio
con:1
```

```
select
    *
from
    realimentacio
order by
    Cerca
```

```

;

[ImportTable]
title:taulaTesaurs
con:1

select
    *
from
    tesaurs
order by
    ID DESC
;

[ImportTable]
title:taulaOpenData
con:1

select
    *
from
    OpenData
;

//Adreces de les icones. Per ordre:
//Cercador; Arbre conductor; SigneMés; SigneMenys; Endarrere
//El número després del nom es refereix als píxels del costat (imatge quadrada)

[Icones]
C:\Users\Josep Solé\Desktop\bin_Cercador\Imatges\panot_60.ico
C:\Users\Josep Solé\Desktop\bin_Cercador\Imatges\tree_view4_37.ico
C:\Users\Josep Solé\Desktop\bin_Cercador\Imatges\plus_20.ico
C:\Users\Josep Solé\Desktop\bin_Cercador\Imatges\minus_20.ico
C:\Users\Josep Solé\Desktop\bin_Cercador\Imatges\back_30.ico

//Les propietats escrites a PropietatsInfo comencen a la tercera columna de la BBDD.
//A la 1a columna hi ha l'índex i a la segona la descripció de la ubi que ens ha dut_
//_fins la lectura de la fila de la BBDD en qüestió.

[TitolFitxa]
Fitxa amb informació detallada de

[PropietatsInfo]
--Tercera columna de la taula taulaInformacio:--
Nom
--Quarta columna de la taula taulaInformacio:--
Adreça
--Cinquena columna de la taula taulaInformacio:--
Correu/Web
--Sisena columna de la taula taulaInformacio:--
Telèfon
--Setena columna de la taula taulaInformacio:--
Responsable/s
--Vuitena columna de la taula taulaInformacio:--
Informació Addicional

```